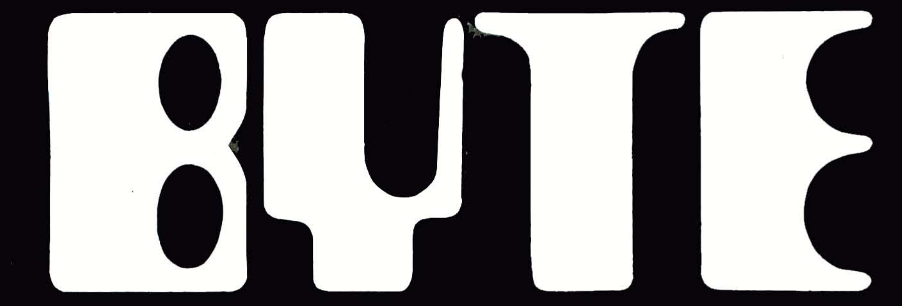
ISSUE NUMBER 10 JUNE 1976 $1.50



PRINTED IN USA

the small systems journal

The Game of LIFE Played in Color

COMPARE

Feature	SWTPC–6800	Theirs—Your Choice
Processor—	The best "Motorola MC6800". Two accumulators, automatic vectoring, seven addressing modes and complete set of branch instruction. The more powerful instruction set and memory orientated architecture makes programming very straight forward and easy to learn. Operates from a single +5 Volt supply.	Some are almost as good in one respect or another. None can offer all of the features of a real MC6800.
Memory—	Static 2102-1 type memories. Fast enough to allow the processor to run at full speed at all times. No refresh cycles, no problems with glitches and flakey dynamic memories.	Various types available. Often not included in the basic kit, and must be purchased as an extra cost option. (this is an option?)
Power Supply—	10 Amp. Capacity. More than enough to power a fully expanded system. Power supply uses a rugged 25 amp bridge rectifier and a 91,000 mfd computer grade filter. Regulators on the individual plug-in cards.	Some expand more than others with the supply provided. Check carefully.
Expansion—	Seven slots for processor and memory boards. Eight I/O slots. I/O's are programmable type. All decoding and clocking provided from mother board making additional interfaces very inexpensive. Baud rates may be independently selected for each interface card.	Varies from "0" to 16, or more.
Start Up—	Automatic start and reset provided by "Motorola" Mikbug® ROM. No fiddling with switches and status lights. Just push the button and go. Use of standard Motorola firmware makes software 100% compatible with Motorola evaluation module programs.	Anything from switch and status light to automatic ROM loading. If ROM is not a standard part, the software may be unique to that machine.
Clock—	Crystal controlled master clock oscillator with high power clock drivers. Insures reliable, consistant operation with no noise problems. Baud rate divider operating from the master clock oscillator provides the various baud rates for the I/O devices with crystal accuracy. No adjustments necessary to lock everything in at the proper frequency.	Anything from cheap dual monostable systems to crystal control. Crystal oscillators are best. Dual one-shots can develop phase overlap problems and are more susceptible to noise problems.
Buffering—	Tri-state bi-directional buffers on all data lines, address lines and clock lines on ALL boards. Insures trouble free noise immune operation.	Various—from full buffering to almost no buffering. Lack of full buffering can lead to noise problems.
Documentation—	Very complete. Our own notebook, plus the "Motorola" Programming Manual and Applications Manual. Detailed instructions along with sample programs help you to understand programming. You will be ready and able to write your own programs after mastering these instructions.	
Price—	$395.00 **For the whole thing.** You get the case, power supply, processor 2K word memory and serial interface. No extras to buy. Just connect a terminal and start operating.	As we said—COMPARE.

® Registered Trademark Motorola



COMING ATTRACTIONS

CASSETTE INTERFACE—Our new AC-30 will make it possible to load and dump programs to cassette tape using ordinary inexpensive recorders. Uses the standard "Kansas City" recording format for compatibility.

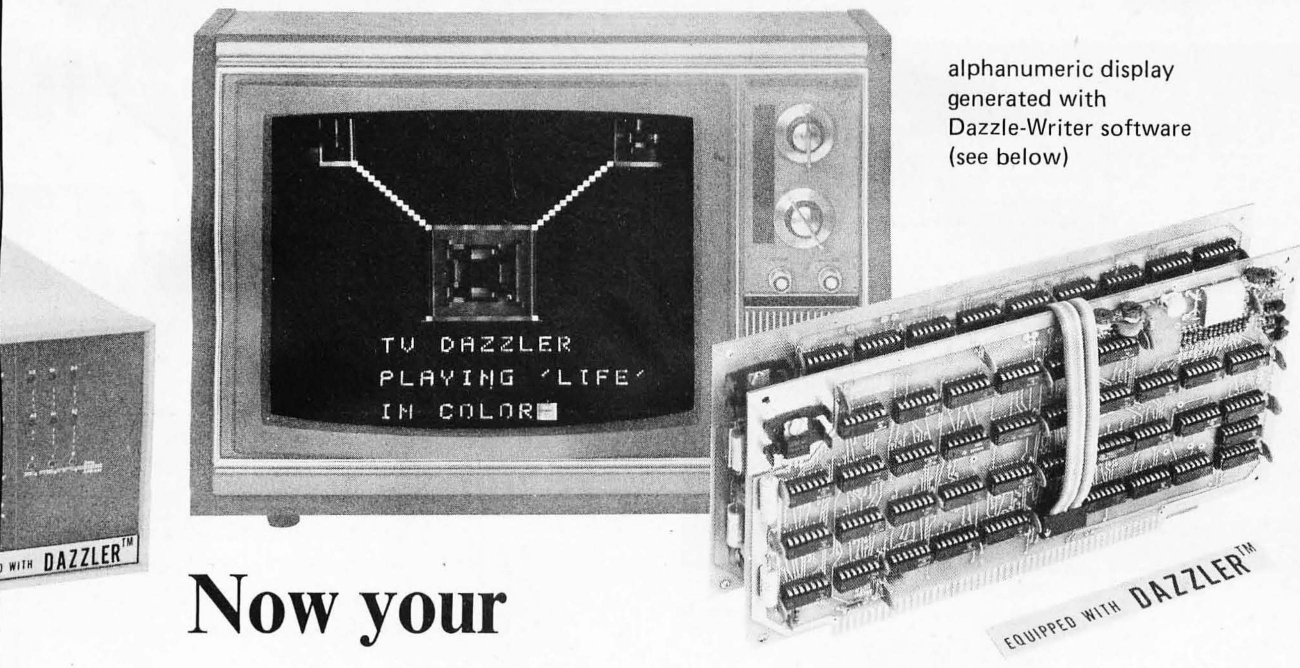
LINE PRINTER—At last—hard copy at a reasonable price. Our new PR-40 printer will print program listings, or any other material you might want to keep in printed form. A dot matrix printer at a price you are not going to believe.

GRAPHICS TERMINAL—A universal—works with any computer—graphics terminal. With this connected to your computer you can play games in style. May be used with our CT-1024 to put both graphics and alphanumerics on the screen simultaneously.

PLEASE—Don't call or write. We will have details on these projects in our next ad. Government regulations prevent us from giving prices, or taking orders yet.

SOFTWARE—The flood is near. Editor and assembler now available. BASIC and more games right away. Yours for the cost of copying. WE DON'T SELL SOFTWARE—WE GIVE IT TO YOU. ENJOY IT, COPY IT, WE WON'T COMPLAIN. . .

SWTPC Southwest Technical Products Corp., Box 32040, San Antonio, Texas 78284



alphanumeric display generated with Dazzle-Writer software (see below)

Now your color TV can be your computer display terminal

New capabilities, too

Cromemco's new computer/tv interface circuit lets you have a full-color computer display terminal for little more than a black-and-white terminal.

The Cromemco interface also lets you do vastly more with your color terminal than you can do with ordinary black-and-whites.

We call our interface the TV Dazzler®. It consists of two circuit boards that plug directly into your Altair 8800 or IMSAI 8080 computer.

Alphanumerics plus action, and graphics

The Dazzler® maps your computer memory content onto your color tv screen in full color.

That doesn't mean just that you see alphanumerics in color. You can display *any* information in memory. And do so in color.

LIFE in color

You can display computer games or animated shows (rocket ships). What's more, you can display business or technical graphics — multi-colored charts, graphs, histograms, educational material — all from computer memory. Even light shows. Not even the biggest computer manufacturers offer all this in color.

Needs only 2K memory

Technically, the Dazzler® scans your computer memory using direct-memory access (DMA). It formats each memory bit into a point on the tv screen to give a 128 x 128-element picture. Only a 2K-byte computer memory is required (only 512 bytes for a 32 x 32 picture). The quality of the pictures is evident in the photos.

The Dazzler® output is a video signal that goes directly to the tv video amp or to the antenna terminal through an inexpensive commercially-available device.

Inexpensive — and so much better

You can see from the list below that the Dazzler® is little if any more in price than an ordinary b/w interface or tv typewriter. But it does so much more.

Order now

By mail or at your computer store

If you're into computers (or want to be), if you want to invent these beautiful displays or games, or to plot colorful material inexpensively at home or in business, the Dazzler® is for you.

Not only is it reasonable, but it's sold at computer stores from coast to coast.

Or order directly by mail on your bank card.

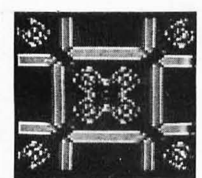
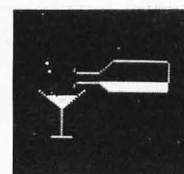
TV DAZZLER® (complete kit) $215
TV DAZZLER® (fully assembled and tested) $350

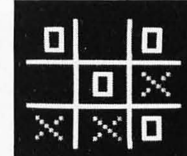
SOFTWARE
(punched paper tape with documentation)

LIFE in full color $15
KALEIDOSCOPE in full color $15
DAZZLE-WRITER (for alphanumeric displays in color) $15
DAZZLE-MATION (for computer-generated animated displays $15
TIC-TAC-TOE (you play the computer) $15

Shipped prepaid if fully paid with order.
California users add 6% sales tax.
Mastercharge and BankAmericard accepted with signed order.
Delivery: from stock for immediate shipment.

TV DAZZLER
SOFTWARE
CONTEST
Write for details

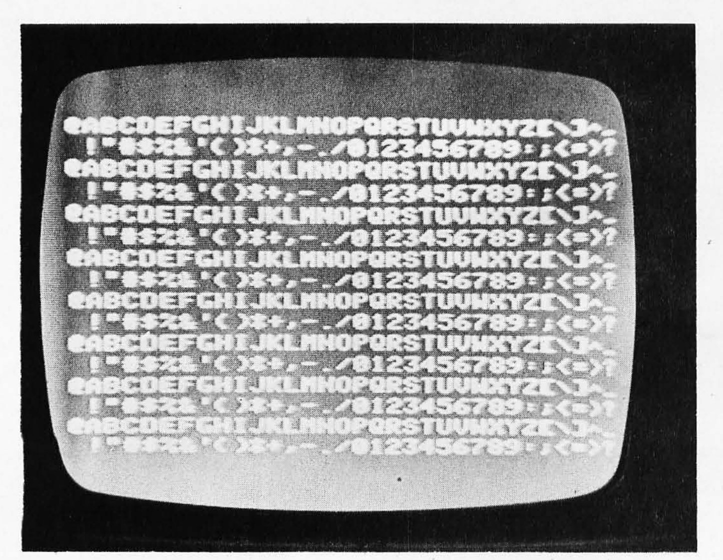




LIFE • DAZZLEMATION • DAZZLE-WRITER • TIC-TAC-TOE • KALEIDOSCOPE
SOME SEQUENCES FROM CROMEMCO SOFTWARE



Cromemco
Specialists in computer peripherals
One First St., Los Altos, CA 94022 • (415) 941-2967

In This BYTE



page 16

What does it take to make your microcomputer keep track of that small spare time "moonlight" business? For small businesses, one of the biggest problems is all the paperwork required of the entrepeneur. John A Lehman provides an introduction to the subject of automated accounting procedures in his description of **A Small Business Accounting System.**

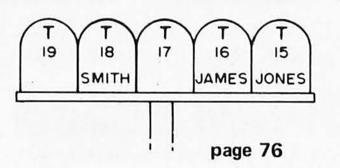
Find out how to **Build a Television Display** which can be interfaced to your computer's memory address space by consulting C W Gantt Jr's article on a 15 IC controller for a 32 by 16 display.

Programming to a large extent is organizing your ideas about what the computer should do. Ronald T Herman gives some basic pointers on **Programming for the Beginner**, concerning the practice of structuring program designs into well defined verbal descriptions, before you generate a single line of code. This practice makes programming easier and less subject to nasty errors which interfere with the goal of a working application.

Is your high school's computing budget crimped? Use Christopher Lett's experience with **A High School Computing System** as a way to show that a small budget does not necessarily rule out getting a system.

A Systems Approach to a Personal Microprocessor is the inaugural article in a new series of detailed design articles which is being prepared for BYTE by Dr Robert Suding. In this first instalment, you'll find some of Dr Suding's views on the philosophy of system design, to lay the groundwork.

Build a computer? Sure. Bob Abbott shows you the circuitry and some photos of a wire wrapped M6800 system in his article on **Building an M6800 Microcomputer.**



page 76

One of the principal uses of a computer is data processing. Phillip L Hansford describes one simple data processing application which he implemented with minimal *equipment:* How to **Strike a MATCH** between penpals for his penpal club.

Last month, Bob Nelson introduced **"Chip" Off the Olde PDP 8/E.** In Part 2, he discusses the interrupt structure, control panel features and support chips of the Intersil IM6100 design.

One key component of system software for any computer is a monitor program. You can find out how to program and **Interact with an ELM** (Eloquent Little Monitor) by reading G H Gable's article on his design of a handy piece of system software.

Techniques of developing the effective address for operands in memory vary with the choice of CPU. In **An Introduction to Addressing Methods.** John Zarrella discusses some of the classical ways computers use to calculate the address of operands in memory. You can use this background information in forming your own opinion about the instruction sets of the various microcomputers presently available.

What's in a language? Nat Wadsworth and Mark Arnold present some information on their interpretive language system, **SCELBAL**, in a product description article.

In the Queue

Foreground

Background

Nucleus

BYTE #10

JUNE 1976

staff

BYTE magazine is published monthly by BYTE Publications, Inc., 70 Main St, Peterborough, New Hampshire 03458. Subscription rates are $12 for one year worldwide. Two years, $22. Three years, $30. Second class postage paid at Peterborough New Hampshire 03458 and at additional mailing offices. Phone 603-924-7217.

Address editorial correspondence to Editor, BYTE, 70 Main St, Peterborough NH 03458.

PERSONAL COMPUTING '76

Consumer Trade Fair

Atlantic City NJ

Personal Computing '76 will be the first large industry wide show for the personal computing field. It is being sponsored by the Southern Counties Amateur Radio Association of New Jersey, and will be held August 28 and 29 1976 at the Shelburne Hotel and convention center on the Boardwalk at Atlantic City. For information on exhibit space and the show in general, contact:

John H Dilks III, chairman
Personal Computing '76
503 W New Jersey Av
Somers Point NJ 08244

Phone: (609) 927-6950.

- For reservations, contact:
  Shelburne Hotel/Motel
  Boardwalk & Michigan Av
  PO Box 1138
  Atlantic City NJ 08404
- Manufacturers who provide products for the personal computing marketplace are invited to attend.
- A series of seminars on subjects related to small computer systems is being arranged.
- The show will be almost entirely devoted to computers and related products.
- Admission to the general public will be $7.50 at the door, $5 if ordered in advance by individuals. Additional discounts are possible for group purchases by clubs, companies and other organizations.

Toward a Parallel Interface Standard

Articles Policy

BYTE is continually seeking quality manuscripts written by individuals who are applying personal systems, or who have knowledge which will prove useful to our readers. Manuscripts should have double spaced typewritten texts with wide margins. Numbering sequences should be maintained separately for figures, tables, photos and listings. Figures and tables should be provided on separate sheets of paper. Photos of technical subjects should be taken with uniform lighting, sharp focus and should be supplied in the form of clear glossy black and white prints (if you do not have access to quality photography, items to be photographed can be shipped to us in many cases). Computer listings should be supplied using the darkest ribbons possible on new (not recycled) blank white computer forms or bond paper. Where possible, we would like authors to supply a short statement about their background and experience.

Articles which are accepted are typically acknowledged with a binder check 4 to 8 weeks after receipt. Honorariums for articles are based upon the technical quality and suitability for BYTE's readership and are typically $15 to $30 per typeset magazine page. We recommend that authors record their name and address information redundantly on materials submitted, and that a return envelope with postage be supplied in the event the article is not accepted. ■

Editorial By Carl Helmers

The idea of a parallel interface standard which encourages interesting combinations of peripherals and processors from different manufacturers is one which in my opinion should be pursued to help foster the growth of the personal computing marketplace, a growth which will provide a wider range of options for both users and suppliers of products.

The User's Eye View

The need of a parallel interface standard from the user's point of view is readily perceived. The parallel interface standard will be the personal computing equivalent of what exists in the audio equipment industry: a widely manufactured, readily available physical interconnection with logically and electrically compatible signal definitions. The plug of choice in the audio field is the RCA style phono plug universally used to interconnect low level audio signals via shielded cables. This enables the purchaser of brand X turntable to plug it into a brand Y receiver using shielded cables of brand Z.

The ideal for the parallel interface definition in personal computing is similar. What is needed is a definition which will allow the owner of brand X processor to plug his system physically, logically and electronically into a brand Y music synthesizer or brand Q graphics display, using brand Z cable assemblies. The interface definition to be created will at a minimum guarantee hardware compatibility. The applications software compatibility can be provided by the peripherals manufacturer in the form of simple relocatable routines with common functional documentation and detail code generation for the various microprocessor instruction sets.

The user will see a much more highly desirable product if it contains provision for the standard interface, since he or she will then be able to interface a wide variety of specialized applications and systems oriented peripherals without the necessity of performing the systems engineering equivalent of reinventing the wheel at the interface level.

The Manufacturer's Eye View

The manufacturer of a product for the personal computing field has as a goal the maximization of sales, and hopefully as a result, the maximization of profits. This is a simplistic economic analysis which ignores the existence of specialized counter examples, but it is generally valid in most cases.

Continued on page 112

Call for Papers

PERSONAL COMPUTING '76 – Technical Session on Standards

As part of the Personal Computing '76 convention August 28–29, Atlantic City NJ, BYTE magazine will coordinate a technical session on areas of standardization for the personal computing field. The purpose of defining standards is to provide a consistent and workable set of specifications for interfaces between different equipment which permit combined operation as a system. Standards are in the interest of manufacturers who will find wider markets as a result, and standards are in the interest of users who will benefit from a wider choice of compatible options.

The following technical areas are identified at this point in time:

Audio Cassette Standard

A provisional interchange standard for audio media was defined by a working meeting sponsored by BYTE magazine in November 1975. This standard will be reviewed, and the idea of a higher performance audio media standard should also be considered.

Parallel Interface Standard

The proposal here is to define a parallel interface standard for the connection of peripherals to central processors. The standard should specify logical interface, electrical interface and possibly a set of preferred physical plug and receptacle designs.

Software Interface Standard

The proposal here is to define several byte sequential media independent record formats for data involved in interchange between systems. This standard will prove useful to the development of software markets, and could be incorporated into the firmware of operating systems and loaders supplied with processors.

Arrangements

Individuals and firms wishing to participate in the discussions should send written position papers on their area of interest to:

Carl Helmers
Editor, BYTE
70 Main St
Peterborough NH 03458
Attention: Technical Standards

Papers should be neatly typed camera ready copy which will be used as is to create a proceedings booklet to be distributed at the technical session. It is suggested that drawings be done using ink or felt tip pen, and that film ribbon be used for typewritten copy.

The standardization categories of audio media, parallel interface definitions, and software data formats should not be viewed as exclusive. Identification of additional areas where standards are appropriate will also help the growth of the personal computing field.

It is expected that positions and proposals presented at the Personal Computing '76 show will be used as a starting point for the second annual BYTE magazine symposium on standards to be held later in the fall. Questions regarding standards activities and coordination of this session should be directed to Carl Helmers at BYTE at the above address, or phone (603) 924-7217.■

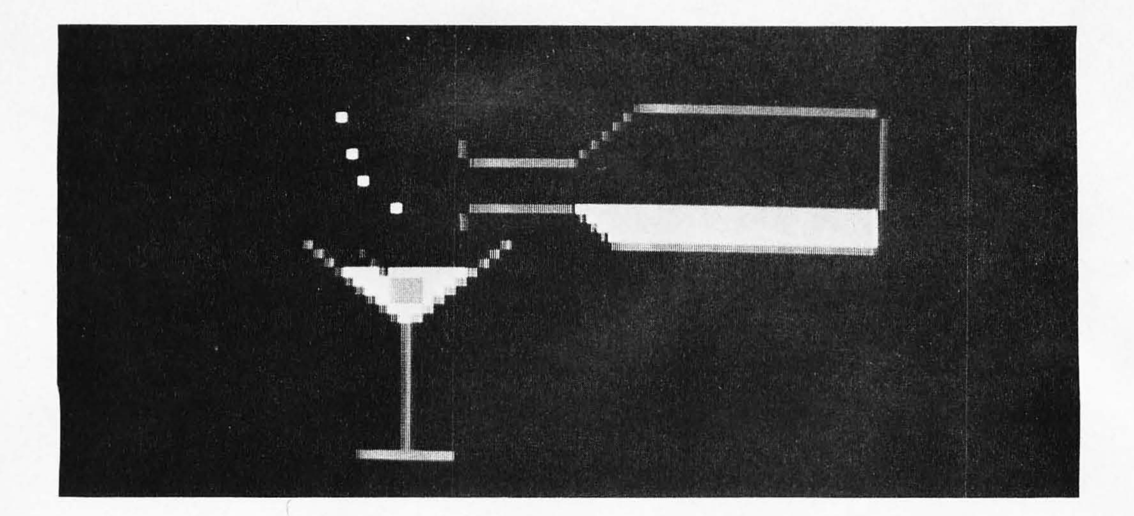


Photo 1: Here is a black and white reproduction of a single frame of a wine pouring animation sequence created by Steve Dompier using his Dazzlemation program. The colors of the original reproduce as shades of grayness in this black and white picture.

About the Cover

Imagine being able to look inside your computer memory, actually being able to see the individual bits. With this sort of X ray vision your computer memory could also serve as your computer display. Messages could be spelled out by lighting some bits and darkening others. Games could be played with clusters of bits forming game pieces and markers. Space War might be played with miniature rocket ship patterns zooming in, out and around the visible region of memory address space. The key element of hardware required to actually achieve this imagined result is a memory module which has provisions to map its contents onto a television screen. This is precisely what Cromemco has done in creating its TV Dazzler product, the results of which were used to create this month's cover.

The TV Dazzler hardware features two modes of operation providing high resolution and low resolution generation of a television picture. Through software selection the TV Dazzler can be programmed either as a 128 x 128 point black and white display, or as a 64 x 64 point colored display. The points of the display grid are tiny square regions on the screen which map into segments of the 2 K byte memory of the TV Dazzler module.

In the high resolution "bit mapped" mode, TV Dazzler uses its 2 K byte memory as a means of storing 2^{14} = 16,382 bits required to generate a unique "on" or "off" value for each location of a 128 x 128 grid. This high resolution black and white mode is very effective for alphanumeric displays and detailed computer controlled images.

In the low resolution "nybble mapped" mode, TV Dazzler uses its 2 K byte memory as a means of storing 2^{12} = 4096 four bit nybbles of data needed to generate a color display on a 64 x 64 grid. Each nybble determines the color and intensity of the corresponding picture element on the grid. The most significant bit sets either high or low intensity, and the next three bits independently select the blue, green and red channels of the color TV signal.

Like a metaphorical beachball, (see January 1976 BYTE editorial), the Dazzler provides the hardware for an incredible variety of applications. This variety is realized through the software for games and other purposes developed by people who buy and use this type of peripheral. One particular application of the peripheral is a program called Dazzlemation which was written by Steve Dompier. The purpose of Dazzlemation is to record an animated sequence of TV frames in color, then play these back. In order to make such a sequence, Dazzlemation is used to color in the appropriate regions of single frames which are stored in memory. Steve's standard demonstration sequence shows a carafe of red wine being poured into a wine glass. One frame of the carafe sequence is illustrated by photo 1. This is just one of an endless variety of computer generated animated displays which is made possible by programs like Dazzlemation.

A second application of the Dazzlemation hardware was used to generate the pattern which forms the main portion of the cover. This is a program called Dazzler-LIFE which was written by Ed Hall. John Conway's

[This short account is based upon materials supplied by Harry Garland of Cromemco. . . . CH]

fascinating game of LIFE gains a new dimension when it is displayed in color. Watching the patterns evolve can be intoxicating in black and white, but becomes truly addictive when color is used to illustrate the game board. In the Dazzler-LIFE program, the game begins in a drawing mode which allows the user to draw an initial colony of cells on the screen using controls from the ASCII keyboard. Then the evolution process is initiated with each succeeding generation being displayed on the screen with colors marking the health of each cell. Cells that are too crowded, or too remote, turn a flaming red color, then wither away. Newborn cells first appear in green, then grow up to a mature blue color. The kaleidoscopic result is fascinating to watch. One frame of a colorful LIFE history was photographed for the cover.

Still another application of the Dazzler is as a hardware game board for sophisticated computer automated games. One example of such an application is the Tic Tac Toe software written by George Tate. Dazzler Tic Tac Toe is written in BASIC, and demonstrates how very well suited the MITS BASIC is for creating colorful creations. George's program is one of a class of "man versus computer" game applications, and is reputed to be extremely competent at Tic Tac Toe. A sample of the output is reproduced here in black and white as photo 2.

A useful utility program for the Dazzler, which demonstrates the bit mapped mode of operation is the Dazzlewriter software created by Ed Hall. This program turns your ASCII keyboard/computer/Dazzler combination into a TV typewriter by generating the 5 x 7 dot matrix display for each keyboard character. A sample of Dazzlewriter activity is shown in photo 3. Since the main memory of the computer is used to store the character generation information, there is no need for any additional hardware beyond the memory requirements of Dazzlewriter.

Another delightful application of the display is an "idling" program you'll probably want to leave in the computer system when you're not using it for another purpose. This program is Li-Chen Wang's colorful Kaleidoscope program. The program is surprisingly short, just 127 bytes long, yet it generates an unending sequence of captivating patterns.

These programs were created by some of the first individuals who had access to the Dazzler hardware. They are written for the 8080 instruction set (except George Tate's BASIC Tic Tac Toe) and are available in paper tape form from Cromemco at $15 each.■

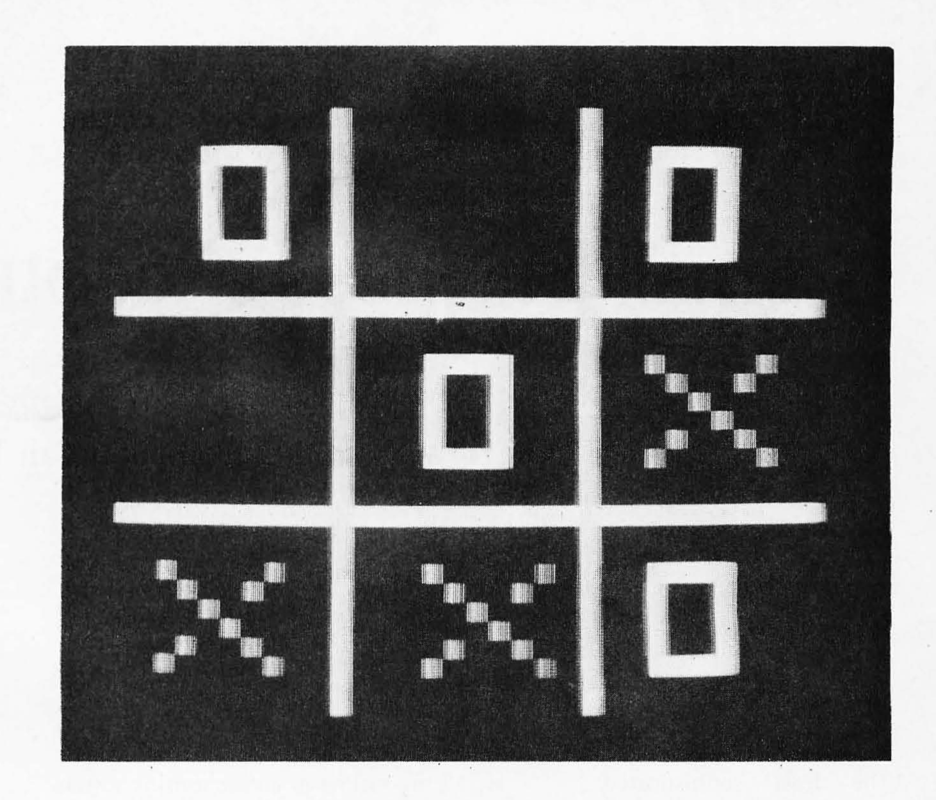


Photo 2: Here is the game board of George Tate's Tic Tac Toe application, written in MITS Altair BASIC with the TV Dazzler as its display peripheral.



Photo 3: Here is a sampling of outputs generated using Ed Hall's Dazzlewriter program to turn the TV Dazzler/computer/keyboard combination into the logical equivalent of a TV typewriter style display.

A Small Business Accounting System

Or, How Your Microcomputer Can Take the Worry Out of Tax Time

John A Lehman
716 Hutchins #2
Ann Arbor MI 48103

The least sophisticated form of bookkeeping is single entry accounting; it is not, however, generally suitable for preparing financial statements for banks, investing brothers-in-law, and so forth.

Double entry bookkeeping has the advantage of incorporating redundancy and error checking techniques. It is the most common form of business accounting.

Here's an outline of an accounting system suitable for small business use on a microcomputer. It is designed for a small, inexpensive system having a central processor, Teletype IO, one or preferably two cassette tapes for storage, and a high level language facility such as BASIC. It could probably be written in assembly language, but at a price of inconvenience. The system is designed to be used by an individual proprietorship (one man business) or a small partnership. While perhaps suitable as a bookkeeping system for a small corporation, it is not intended to produce the sort of reports which various regulatory agencies may require of one. It is designed to keep books, produce tax returns (either Form 1040 schedule C for proprietorships or Form 1065 for partnerships), produce balance sheets which may be required either for management information or for the information of banks and other outside investors, and to be adaptable for check reconciliation, cash budgets, pro forma balance sheets and the like. Its use requires about the same amount of time and effort as keeping a journal would normally, with the added advantage that the entries are pretty much self checking. All other reports are produced by the programs which would be used. I'll try to describe the system in enough detail so that anyone who is skilled in BASIC and knows a little about accounting could write a program to do all of the above.

First, however, it might be a good idea to take a quick look at accounting systems and what they're used for.

Of the various systems available, the simplest is the single entry system. A check book is a good example; each time money goes in or out, a notation is made of the date, the amount, and any comments on sources, uses, etc. This sort of system is obviously very simple to keep, and has the additional advantage of being accepted by the IRS for preparing tax returns. However, it has a number of disadvantages. The first is that it is not self checking, as anyone who has ever tried to balance a checkbook can testify. Also, while capable of producing an "income statement" (the generic term for what a tax return amounts to), it is not suitable for the preparation of other financial statements that may be required by banks, investing brothers-in-law and so forth. These disadvantages make a single entry accounting system unsuitable for the system under discussion here.

Double Entry Accounting

The other major accounting system is the double entry system. It was invented about 600 years ago, and came into widespread use because it was self checking. It is also quite a bit more complicated than a single entry system. The basic idea behind the double entry system is that each transaction has two parts: where money comes from and where it goes. So each transaction is entered twice, each time in a different account. The mechanism behind this is the idea of debits (DR) and credits (CR). Very briefly, a debit represents an addition to something which you have (an asset) or to an expense. A credit represents a subtraction from one of these. On the other hand, a debit represents a subtraction from something which you owe or from a revenue, while a credit represents an addition to one of these. All of which can be very confusing.

As a quick example, suppose you pay

$100 on your BankAmericard and receive $150 for some service which you performed. You would debit accounts payable (subtracting from what you owe) for $100, and credit cash (subtracting from something you have) for $100. Then you would debit cash (adding to something you have) for $150, and credit income or revenue (adding to revenue) for $150. The self checking feature is provided by the fact that debits must always equal credits. It would probably be a good idea to look through a beginning accounting book to get more examples to help explain accounting techniques. I've listed some at the end of the article.

Besides being self checking, a double entry system has the advantage of being able to churn out all sorts of reports on what is going on in the business in question. The IRS approves of it; and in fact, large companies have no choice – they *have* to use it. Now that we've described the major accounting systems, let's get on to what they do and how they can be used in a computerized system.

The purpose of any accounting system is to provide information (another purpose is to provide employment for accountants, of course). This information is of use to various people. The owner of a business uses it to see how well he's doing, and more important, where he's not doing so well. Another important user is your friendly local IRS agent; anyone in business is required to produce accounting reports to the Internal Revenue Service's specifications. Banks and other investors also are likely to be quite interested in this sort of information, especially when their services are requested for loans rather than for deposits. Corporations are also required to provide statements to various government agencies, but we're not going to be concerned with that here.

The basic statements and reports were mentioned earlier. The first is the balance sheet such as the one shown in figure 1. This represents the financial state of the company at a particular time. The left hand side (in the US at least) represents assets, or what the firm has. The right hand side represents liabilities and equities. (Liabilities and Equities is accounting terminology for where the stuff on the left came from.) Liabilities are amounts owed; equities are amounts contributed or earned by the owner(s). The second statement is the income statement. As was mentioned above, a tax return is a species of income statement. This shows what happened over a period of time. Other statements, such as the cash budget and the pro forma balance sheet, show what may happen in the future. These are the statements which our system is going to be able to churn out. Now, having got an overview of what we're trying to do, let's take a look at our data base requirements.

The first thing we are going to need here is a set of names for our accounts. This is a "chart of accounts" to use the jargon of the accounting trade. A small system such as ours will need about 35 of these, selected for the most part to make our output match

ASSETS:		**LIABILITIES:**	
Cash	1000	Payables	2000
Receivables	2000	Notes from bank	1000
Equipment	4000		
		EQUITY:	
Total	7000	Proprietor	4000
		Total	7000

Figure 1: The Balance Sheet. This document shows the current financial state of a business operation. It is used by businesses large and small, and is one of the end products of the automated accounting system.

EXPENSES (Debit to add, Credit to subtract)	
Return & Allowances	RTN
Depreciation*	DEP
Business Taxes*	TAX
Rent	RNT
Repairs*	RPR
Salaries & Wages	SAL
Insurance	INS
Professional fees	PRF
Commissions	COM
Amortization*	AMT
Pension/Profit sharing	PEN
Interest	INT
Bad Debts	BDB
Depletion	DPL
Other (specify if common, eg:	MIS
Fuel	FUL
Electricity	PWR
Telephone	FON
Cost of Goods Sold which includes	CGS
Purchases	PUR
Materials/supplies	MAT
Other costs	OTR
Labor (used for or directly related to Production – does not include money paid to you)	LAB

ASSETS (Debit to add, Credit to subtract)	
Cash	CSH
Receivables	RBL
Inventory	INV
Prepaid expenses	PPD
Supplies	SUP
Equipment	EQT
Investments	IVS
Misc.	ETC

LIABILITIES & EQUITY (Credit to add, debit to subtract)	
Payables	PBL
Notes	NOT
Long Term Payables	LTP
Proprietor	PRP
Drawing	DRW

REVENUES (Credit to add, Debit to subtract)	
Gross Receipts	RCP
Other Revenue	REV

***Items for which the IRS requires supplementary schedules or statements**

Figure 2: Account Files Example. When the double entry accounting system is designed, one of the first steps is to create a list of accounts and their corresponding mnemonic codes. The mnemonic codes are used internally by the computer in order to save memory space. If you are lavish with memory, texts of the long names could be looked up in a table when you generate reports.

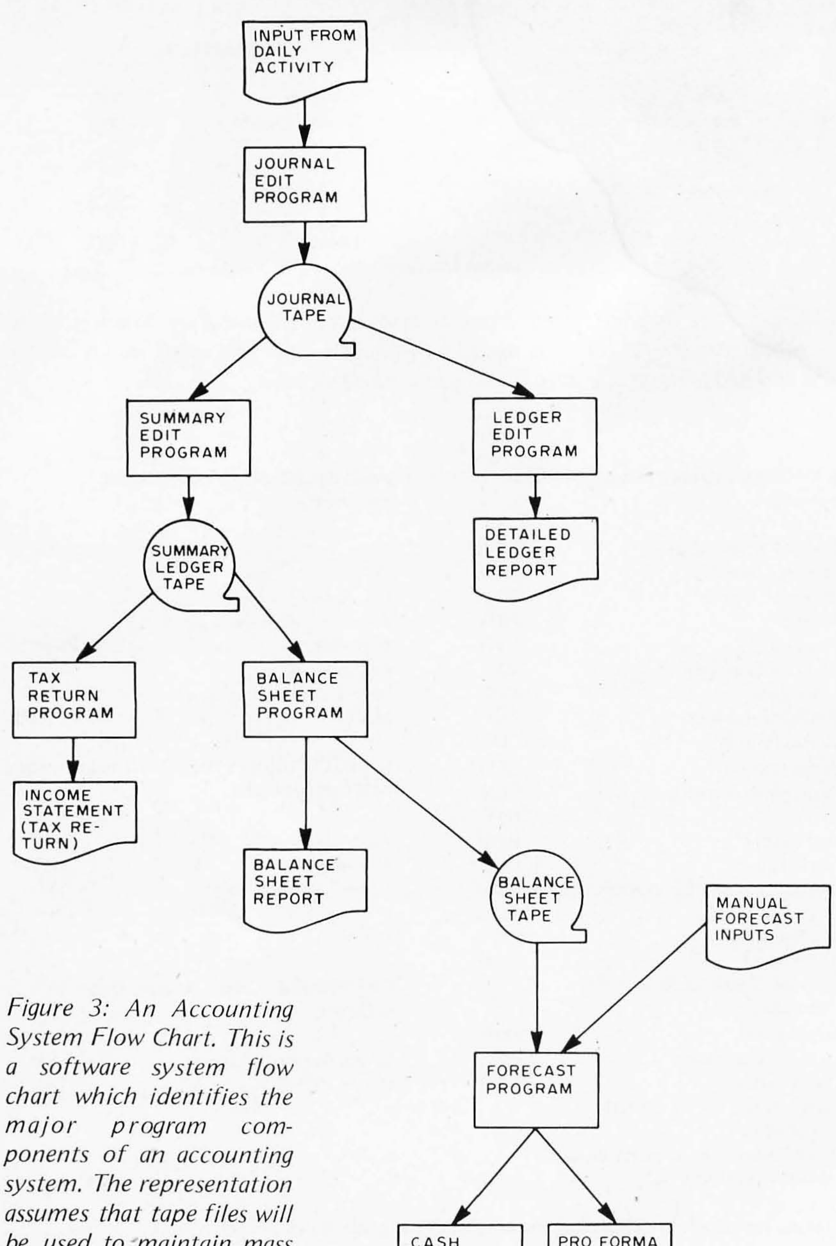


Figure 3: An Accounting System Flow Chart. This is a software system flow chart which identifies the major program components of an accounting system. The representation assumes that tape files will be used to maintain mass storage on your personal system; floppy disks with sequential access file organizations could be used as well.

what the IRS requires. In order to save memory space in the programs, each of these accounts is also given a three letter mnemonic code. Two letters would be possible, but some ease of use would be sacrificed. Figure 2 gives a sample list of accounts and mnemonics, broken down by classification. A brief explanation of some of the accounts might be in order. Returns and Allowances is for goods which are returned for one reason or another. Its purpose is to reduce the amount in gross receipts while keeping the amount of returns separate. The category SAL includes only those wages paid which are not included in cost of goods sold (CGS). This would involve such things as clerical help. INT is interest paid, not received. BDB (bad debts) is used if we want to use the specific charge off method of accounting for such unfortunate happenings. The IRS also allows use of another method, called the reserve method, but it is more complicated. DPL (depletion) is used for things like oil wells and mines. DEP (depreciation) is used for equipment, machines and the like, while AMT (amortization) is used to charge part of the cost of such things as organization expense, capitalized research and development and so forth. Some of these things can be listed as assets when the money is first spent, and the cost spread over several periods. For details see the IRS books listed at the end of the article. Cost of goods sold (CGS) is the total of the costs incurred to get something ready for sale; the breakdown is listed below it. Cash (CSH) is mostly checking account balances. Receivables (RBL) are what customers owe you on account. Payables (PBL) are what you owe on account. Proprietor (PRP) is what you put into the business and what it has earned so far. Drawing (DRW) is the account you use to take money out of the business for personal uses. Notes (NOT) is money borrowed from banks and other lenders. The rest should be pretty much self explanatory. These 35 or so accounts are the data files which we're going to be working from; all of the information we put into the system goes into them and all of the output uses them as building blocks. Now, having taken care of all of the groundwork, we are ready to start running information through the system.

Reference to the system flow chart of figure 3 shows that the journal is the first thing we produce. It's shown being produced on tape, since that way we can use it to produce all of the other reports without having to type in any more material, at least until we come to the forecasts. Also, by writing our journal entries onto tape as soon as they're checked by the editing program, we save much memory space, since we need keep only a little bit of data in memory at any given time. So, in this automated system, the journal is the only file we really have to manipulate on a day by day basis. To use it, first we enter the date. Then we enter each transaction through the checking program which makes sure we have two entries for each amount and that the numbers we give the machine match. A sample of a possible format is given as figure 4. We debit the power expense account (remember, we debit an expense when we want to add to it) for $58, and enter the comment that this is for the month of March. Then we credit cash (to decrease it), but reverse the

numbers. The program sees that the debits do not equal the credits, and fires off an error message, prompting us to enter a correction. Note here that we include the check number; this is very important when it comes time to reconcile our records with what the bank statement says. Also, the editing program should provide the ability to debit and credit unequal numbers of accounts so long as the totals are equal. If this would be too much of a demand on memory, amounts can be split up before entry. Going on, the OK indicates that the entries check, and at this point they should be written onto the tape. Entries for the journal can come from cash register tapes, bills, etc. Up through this point our system is about as much work as a manual system, but from here on in things get much easier.

The next item on the system flow chart is the ledger. This is a set of files which puts all of the journal entries for each account together. In our system, there are two types: summary and detailed. In a more advanced system, all of the ledgers would be detailed, but this would require much more memory than most small systems would have available. Basically, what we do at this point is have the program read the journal entries one by one and keep a running count of the amount for each of the different accounts in use. Beginning balances may be read in either via the Teletype or via a separate ledger tape. The ending balances should be printed on the Teletype if the user wishes to see what they are, but they should also be saved on tape for use in preparing the rest of the statements. Detailed ledgers will require a separate run for each one desired; they might be run on a weekly or monthly basis. The most important one is the cash ledger, since this will provide a record of every check written and every deposit made to the checking account by date and number. This should make balancing one's checkbook a fairly simple task. The one thing to be careful of in this program is to be sure that the rules for addition and subtraction of debits and credits are carefully written into the program. Otherwise all that will come out is garbage.

Once we have the ledger, it's fairly easy to see how the balance sheet is generated. A look back at figure 1 will show that there are only about a dozen of the ledger accounts which have to be put together. All of the asset accounts are added together, and the sum is listed at the bottom of the column as total. Subtracting the sum of the liabilities from the sum of the assets leaves what is left for the owner. If the amount in the drawing account is set beforehand, that leaves only the Proprietor (PRP) account to be "plugged," which is to say, given whatever value is necessary to make the two columns come out equal. So, if the assets total $7000, the liabilities total $3000 and there are $500 in the drawing account, that leaves 7000–3000–500=3500 for PRP. The only other detail is that the program should either write the date at the top, or it should be filled in by hand. A balance sheet may be prepared at any time; it will often be required for getting a loan from a bank. Besides being run on paper, it should be run onto tape for use in preparing forecasts.

Probably the most important report which our system will prepare is the income statement. This is a report which shows what has happened over a period; usually a year, but often prepared on a quarterly or a monthly basis. Its importance arises not so much from the fact that people like to see how much money they've made as from the fact that the government is quite interested in this information – so they can take their cut, of course. The system being illustrated produces an income statement patterned

The balance sheet is a snapshot of the current status of the business.

A mass storage file comes in handy for business accounting, since much of the work involved is accomplished by reviewing the same data with different criteria to produce reports.

Figure 4: An Example of the Interactive Dialog with the Journal Edit Program. The purpose of this program is to filter your own manual inputs looking for certain known discrepancies which can be detected by the double entry bookkeeping method. In this example, upper case letters are the computer output to a Teletype (or video terminal) and the lower case letters indicate manual keyboard inputs taken from daily activity records such as receipts, checks written, etc.

Interactive program for journal entries might read:

```
ENTER NAME OF ACCOUNT DEBITED,AMOUNT,AND COMMENTS SEPARATED BY COMMAS:
pwr,58,march
ENTER ACCOUNT CREDITED,AMOUNT,AND COMMENTS SEPARATED BY COMMAS:
csh,85,check 346
DEBITS DO NOT EQUAL CREDITS—ENTER IF DR OR CR TO BE CHANGED:
cr
ENTER ACCOUNT CREDITED,AMOUNT,AND COMMENTS SEPARATED BY COMMAS:
csh,58,check 346
OK
ENTER NAME OF ACCOUNT DEBITED,AMOUNT,AND COMMENTS SEPARATED BY COMMAS:
iam done
OK GOODBYE
```

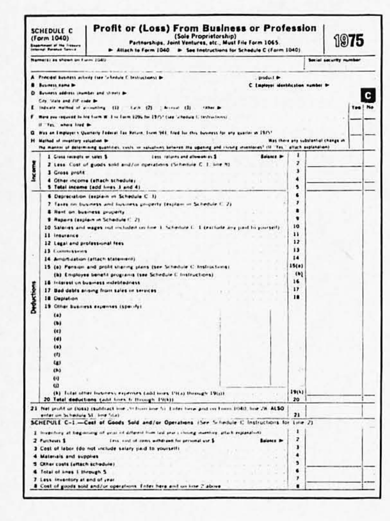
after Form 1040 Schedule C (figure 5), but could produce Form 1065 for partnerships with minor changes. As is fairly obvious to those who can wade their way through the governmentese, what we have to do here is state all income and then subtract expenses. The accounts which we have been working with will do this on what is called an accrual basis, which is to say future expenses and revenues are included if they are certain and we know how much money is involved. For example, if we have charge customers, we include what they are scheduled to pay us in revenues. For a small business it is often better to file a tax return on the cash basis in which only cash in is considered revenue and cash out is considered expense. This system can prepare cash basis returns too; one must eliminate receivables, payables, prepaid expenses and materials and supplies not yet part of cost of goods sold. The effect of all of these should be taken out of the revenue and expense accounts too.

That's the basic system. Using this system alone would be a pretty respectable accounting setup for a small business. But as long as we're using a personal microcomputer, we might think of adding a few bells and whistles. These would pretty much depend on individual wants. We could have the computer automatically calculate FICA deductions when payroll expense is debited. We might also have the machine figure our depreciation and amortization schedules for us. For this we would need (for each item or class of items) initial value, estimated life and age. For tax purposes we would want to get our annual depreciation by taking two divided by the life of the object and multiplying the total times the remaining value. In more symbolic form:

(2/total life)*(initial – depreciation).

This would give us the depreciation to date and the amount for this year, both of which are needed for the flip side of the tax form. We could also do forecasting with the system. For this we would want an interactive program which would ask for estimated expenses and receipts in all the different categories for x number of months. Then we would prepare a (pro forma) balance sheet for the end of the period if our predictions were correct, so that we could see where things would stand if the predictions came true. It could also prepare a month by month schedule to show whether the firm would have enough on hand to meet projected outflows. This is called a cash budget, and is quite a handy thing to have since it enables you to forecast cash shortages far enough in advance to do something about them, and also to compare the results of different courses of action.

And there's the system. While not very fancy from either an accountant's or a system designer's point of view, it ought to be enough to handle much of the recordkeeping for those firms on the other end of the spectrum from GM, IBM and ITT. It might be too that the availability of a few business oriented systems like this will help increase the sales of microcomputers and bring the prices down even more through mass production.■



SCHEDULE C (Form 1040)

Profit or (Loss) From Business or Profession
(Sole Proprietorship)
Partnerships, Joint Ventures, etc., Must File Form 1065.
► Attach to Form 1040 ► See Instructions for Schedule C (Form 1040)

1975

Name(s) as shown on Form 1040 | Social security number

A Principal business activity (see Schedule C Instructions) ► ; product ►
B Business name ► C Employer identification number ►
D Business address (number and street) ►
City, State and ZIP code ►
E Indicate method of accounting: (1) Cash (2) Accrual (3) Other ►
F Were you required to file Form W-[illegible] for 1975? (see Schedule C Instructions) — Yes / No
If "Yes," where filed ►
G Was an Employer's Quarterly Federal Tax Return, Form 941, filed for this business for any quarter in 1975?
H Method of inventory valuation ► Was there any substantial change in the manner of determining quantities, costs, or valuations between the opening and closing inventories? (If "Yes," attach explanation)

Income
1 Gross receipts or sales $ ... Less: returns and allowances $ ... Balance ► 1
2 Less: Cost of goods sold and/or operations (Schedule C-1, line 8) 2
3 Gross profit 3
4 Other income (attach schedule) 4
5 Total income (add lines 3 and 4) 5

Deductions
6 Depreciation (explain in Schedule C-3) 6
7 Taxes on business and business property (explain in Schedule C-2) 7
8 Rent on business property 8
9 Repairs (explain in Schedule C-2) 9
10 Salaries and wages not included on line 3, Schedule C-1 (exclude any paid to yourself) 10
11 Insurance 11
12 Legal and professional fees 12
13 Commissions 13
14 Amortization (attach statement) 14
15 (a) Pension and profit sharing plans (see Schedule C Instructions) 15(a)
(b) Employee benefit programs (see Schedule C Instructions) (b)
16 Interest on business indebtedness 16
17 Bad debts arising from sales or services 17
18 Depletion 18
19 Other business expenses (specify):
(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)
(k) Total other business expenses (add lines 19(a) through 19(j)) 19(k)
20 Total deductions (add lines 6 through 19(k)) 20
21 Net profit or (loss) (subtract line 20 from line 5). Enter here and on Form 1040, line 28. ALSO enter on Schedule SE, line 5(a) 21

SCHEDULE C-1.—Cost of Goods Sold and/or Operations (See Schedule C Instructions for Line 2)
1 Inventory at beginning of year (if different from last year's closing inventory, attach explanation) 1
2 Purchases $ ... Less: cost of items withdrawn for personal use $ ... Balance ► 2
3 Cost of labor (do not include salary paid to yourself) 3
4 Materials and supplies 4
5 Other costs (attach schedule) 5
6 Total of lines 1 through 5 6
7 Less: Inventory at end of year 7
8 Cost of goods sold and/or operations. Enter here and on line 2 above 8

Figure 5: The object of much of this program activity is filling out IRS Schedule C for your small business.

GLOSSARY

Accrual: Including payments and receipts in the future.

Check reconciliation: Accounting buzzword for balancing a checkbook.

Credit (CR): An addition to the righthand side of the balance sheet or to income.

Debit (DR): An addition to the lefthand side of the balance sheet or to an expense.

Journal: The accounting equivalent of a check register.

Ledger: Book or file which contains the totals from the journal broken down by categories.

Payables: Amounts which will have to be paid in the future.

Pro forma: Buzzword used to describe reports which show how things might be or might have been rather than what they are.

Proprietorship: A one man business; one owner.

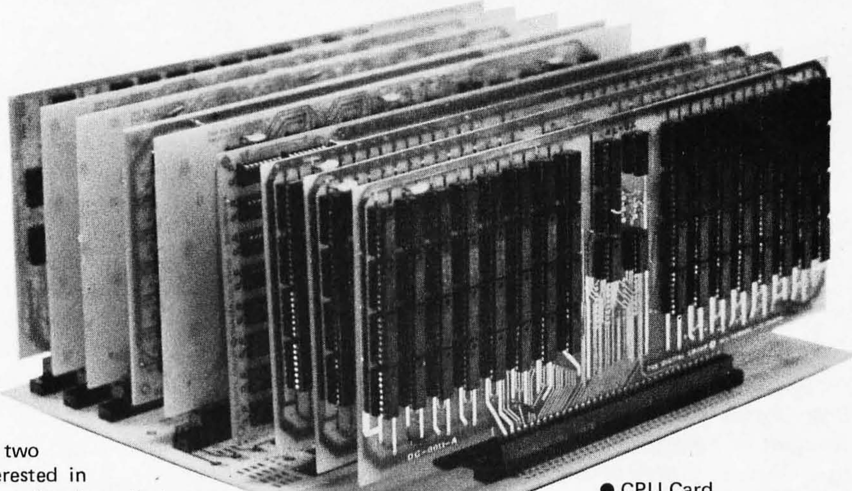
Receivables: Amounts which are not yet on hand in cash but which will definitely be coming in in the near future.

REFERENCES

1. *Accounting Essentials*, Margolis, Wiley and Sons 1972.
2. *Elementary Accounting*, College Outline Series #39.
3. *Management Accounting*, Anthony and Reece, Irwin, Inc, 1975 (Note: this is a college accounting textbook – heavy reading).
4. *Recordkeeping for a Small Business*, IRS #583, 1976.
5. *Tax Guide for Small Business*, IRS #334, 1976 (complete handbook).
6. *Tax Information on Accounting Periods and Methods*, IRS #538, 1975.

The last three are available free from any IRS office.

MEET THE DIGITAL GROUP



If you're taking time to read this ad, we already know a thing or two about you. We know you're interested in devouring everything you can get your hands on that deals with microcomputer systems, and we also figure you're out there shopping around . . . looking for the most sensible products to fit your needs and your budget.

That's why we think you should get to know us.

We're the Digital Group, a relatively small, 18-month-old organization obstinately dedicated to providing quality in every product we offer. You may have already heard a little something about us from a friend — our reputation does seem to be getting around quickly, even though we've never advertised before.

We think it's due to a number of important factors: state-of-the-art designs, a really complete systems philosophy, unexcelled quality, reasonable software, three-week delivery, and no pre-announcements until we're ready to deliver. Our products are not just a gleam in our designers' eyes; they are currently being delivered . . . fast!

The Advantages

Here are a few specific advantages of our product line:

- We offer CPU's from different manufacturers which are interchangeable at the CPU card level. That way, your system won't become instantly obsolete with each new design breakthrough. The major portion of your investment in memory and I/O is protected.
- Digital Group systems are complete and fully featured, so there's no need to purchase bits and pieces from different manufacturers. We have everything you need, but almost any other equipment can be easily supported, too, thanks to the universal nature of our systems.
- Our systems are specifically designed to be easy to use. With our combination of TV, keyboard, and cassette recorder, you have a system that is quick, quiet, and inexpensive. To get going merely power on, load cassette and go!
- Design shortcuts have been avoided — all CPU's run at full maximum rated speed.

The Features

Digital Group Systems — CPU's currently being delivered:
8080A/9080A 6800 6500 by MOS Technology

All are completely interchangeable at the CPU card level. Standard features with all systems:

- Video-based operating system
- Video/Cassette Interface Card
  - 512 character upper & lower case video interface
  - 100 character/second audio cassette interface
- CPU Card
  - 2K RAM, Direct Memory Access (DMA)
  - Vectored Interrupts
  - 256 byte 1702A bootstrap loader
  - All buffering, CPU dependencies, and housekeeping circuitry
- Input/Output Card
  - Four 8-bit parallel Input ports
  - Four 8-bit parallel Output ports
- Motherboard

Prices for standard systems as featured above, start at $425 for 8080 or 6800 and $375 for 6500.

More

Many options, peripherals, expansion capabilities, and accessories are already available. They include rapid computer-controlled cassette drives for mass storage, color graphics, memory, I/O, monitors, multiple power supplies, prototyping cards, and others. Software packages include Tiny BASIC Extended, games, ham radio applications, software training cassettes, system packages, and more (even biorhythm).

Sounds neat — now what?

Now that you know a little about who we are and what we're doing, we need to know more about you. In order for us to get more information to you, please take a few seconds and fill in our mailing list coupon. We think you'll be pleased with what you get back.



THE DIGITAL GROUP, Inc.
P.O. Box 6528
Denver, Colorado 80206
(303) 861-1686

OK, I'd like to get to know you guys better. Send me the whole package!

Name ______________________

Address ______________________

City/State/Zip ______________________

What's New?

16 Bits Doth a General Purpose Minicomputer Make

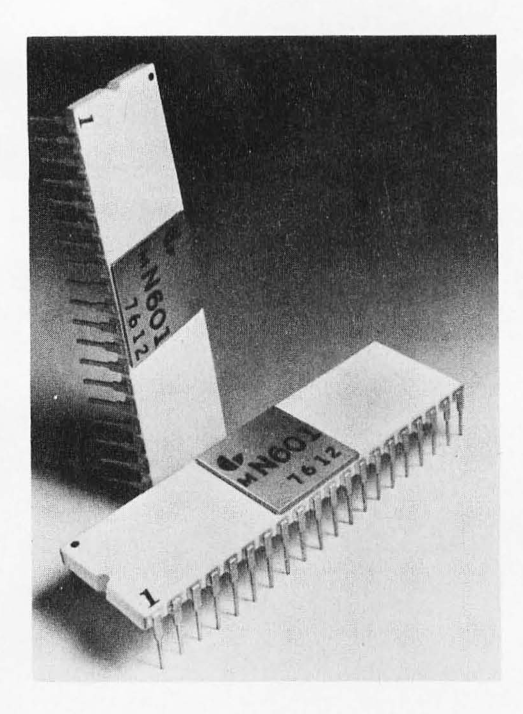


Photo 1: Here is a pair of the new mN601 microNOVA chips which come in 40 pin dual in line ceramic packages. The performance of the microNOVA processor is quite good as microcomputers go. The 16 bit accumulator to accumulator addition time is quoted as 2.4 µs, and the accumulator load time is quoted as 2.9 µs. This compares very favorably with the original Data General NOVA minicomputer performance of several years ago.

A new option in the high performance 16 bit microcomputer field has now arrived on the scene. This is the Data General microNOVA, introduced early in March as BYTE's June issue went to the press. Here is another instance of a widely available minicomputer architecture being made available in the form of a relatively inexpensive microcomputer.

For those who are new to the computer field, the Data General NOVA line of minicomputers is one of the oldest and most widely used products in the conventional minicomputer product area. NOVAs are available with performance which now ranges from the microcomputer chip level to the complete general purpose data processing systems represented by the top of the line ECLIPSE computer. NOVAs have long been an option used by persons who purchased minicomputers as combination business/pleasure computers, and the new microNOVA product will expand this area of use. Based on the product description information supplied by Data General, here is a summary of the microNOVA product's semiconductor line:

mN601: 40 pin package NMOS central processor. The full NOVA instruction architecture is implemented in a single chip design, which also includes several additional features not generally found in NOVA computers:

- hardware stack and frame pointer
- SAVE and RETURN instructions for efficient subroutine linkages
- 16 bit hardware multiply and divide
- real time clock
- memory management logic: memory

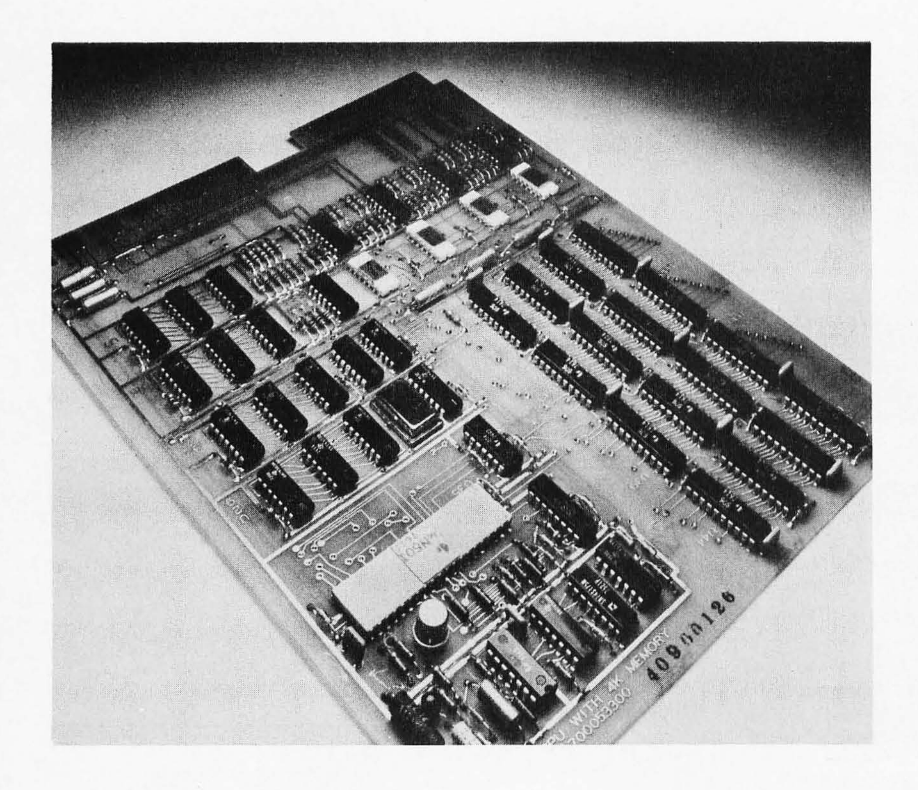


Photo 2: One of the simplest systems products above the chip set level is the microNOVA computer on a board product, Model 8563. This version of the microNOVA one board computer includes 4 K words of semiconductor memory and the ability to run software accumulated during Data General's eight year history with the NOVA computer line.

control, timing and hidden refresh logic for dynamic RAMs.

- device data channel control

mN603: 40 pin package, NMOS, IO controller. This is an integrated circuit designed to implement the standard NOVA IO *instruction set via* DMA operations.

System Buffer Elements: A family of several integrated circuits designed to provide system expansion capabilities.

Data General provides a fully integrated line of hardware packaging and software support to accompany this microNOVA system. Data General primarily orients its products towards the OEM and systems integration markets, so this product will probably prove most attractive to a fairly sophisticated personal computing user, or to the person who works with OEM computing as part of his or her professional activities. But for those who know how to specify and purchase a minicomputer system, a floppy disk based microNOVA system with BASIC or FORTRAN capabilities could prove to be a quite powerful personal computing system useful for both business and pleasure pursuits.

Data General Corporation is located in Southboro MA 01772. Sales and service offices are located in major cities.■



Photo 3: For the do it yourself systems house, Data General provides a complete line of modular components which can be integrated to form a microNOVA system. One of the most interesting characteristics of the microNOVAs integrated to the minicomputer level is a "Programmer's Helper" remote console packaged in a calculator style case at the end of a ribbon cable. The console is shown in its storage position in the front panel assembly at the center of this picture.

Build a Television Display

C W Gantt Jr
6 Fieldpoint Rd
Aurora IL 60538

As a small system expands and becomes more sophisticated, the limiting factor is often the speed of input and output (IO). In addition to being noisy, mechanical, and paper consuming, the slow clacking of a TTY may account for a large percentage of system time. Among the alternatives, the display of characters on a standard TV set is among the simplest and most economical methods.

This TV display (TVD) is designed to take data from 512 bytes of memory and convert it into a video signal with 16 lines of 32 characters. This can be used to feed a black and white or color TV. The data in the TVD memory is in a six bit ASCII subset and is updated by the CPU to create the desired display. The processor addresses the TVD memory just as it does any other portion of memory and can actually execute instructions from the TVD memory if so programmed. Of course, some provisions must be made to prevent the CPU and TVD from simultaneously accessing the TVD memory (more about this below).

As designed, the TVD is strictly a display device with the central processor of your system doing all housekeeping (entering characters, etc). This approach simplifies the hardware at the expense of extra software, but also allows the user to take advantage of the flexibility offered by software data manipulation and formatting.

At present one TVD is up and running in my system, but the memory and central processor interfaces are incomplete. The remainder of this article therefore emphasizes the TVD design and only offers some basic ideas on interfacing to processors. Although simple items such as the power supply and oscillators have been omitted, the information furnished should be sufficient for the more experienced readers to assemble a working version. The straight forward TVD design allows easy modification to meet individual system requirements.

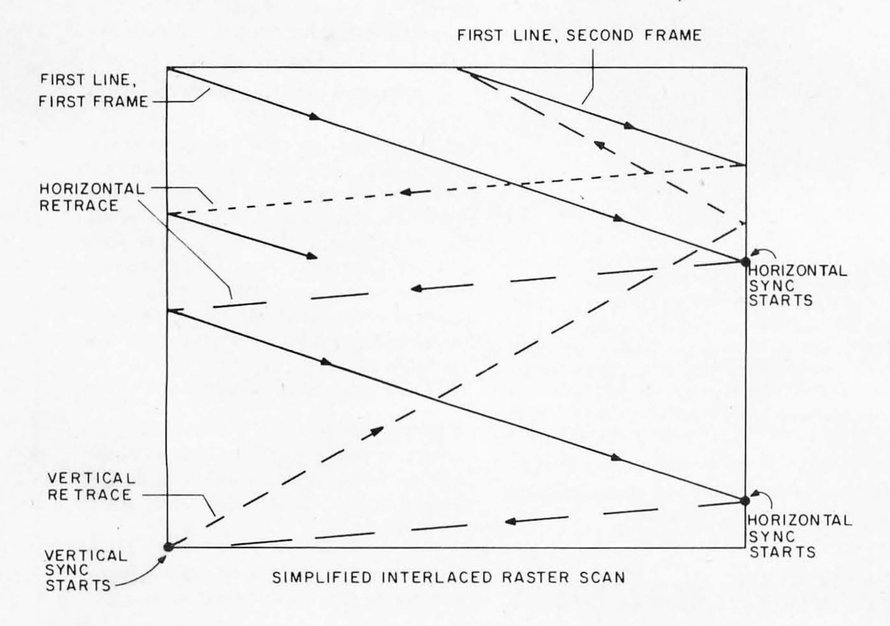


Figure 1: This shows how the electron beam is moved during an interlaced scan in a television monitor. The dashed lines are quick retrace motions which are normally invisible. The solid lines are periods during which the display presents video information controlling brightness on the tube face.

Television Raster Scanning

Before going too deeply into operation of the TVD, a review of the basic television scanning system will clarify some terms with which pure digital designers may not be familiar.

A television picture is formed by scanning an electron beam across the face of the picture tube. A TV line is one sweep of the electron beam from the left of the picture tube to the right (as viewed from the front of the set) and is initiated by the horizontal sync (see figure 1). The horizontal sync pulse causes termination of a line, horizontal retrace of the electron beam back to the left side of the screen, and the start of a new line. During the time of retrace the beam is blanked so that the retrace will not be seen. The time allotted for each complete line (including retrace) is 63.5 microseconds. Of this about 16% is taken by retrace, leaving 53.5 μs of usable line. Video information in the form of a voltage fed to the picture tube

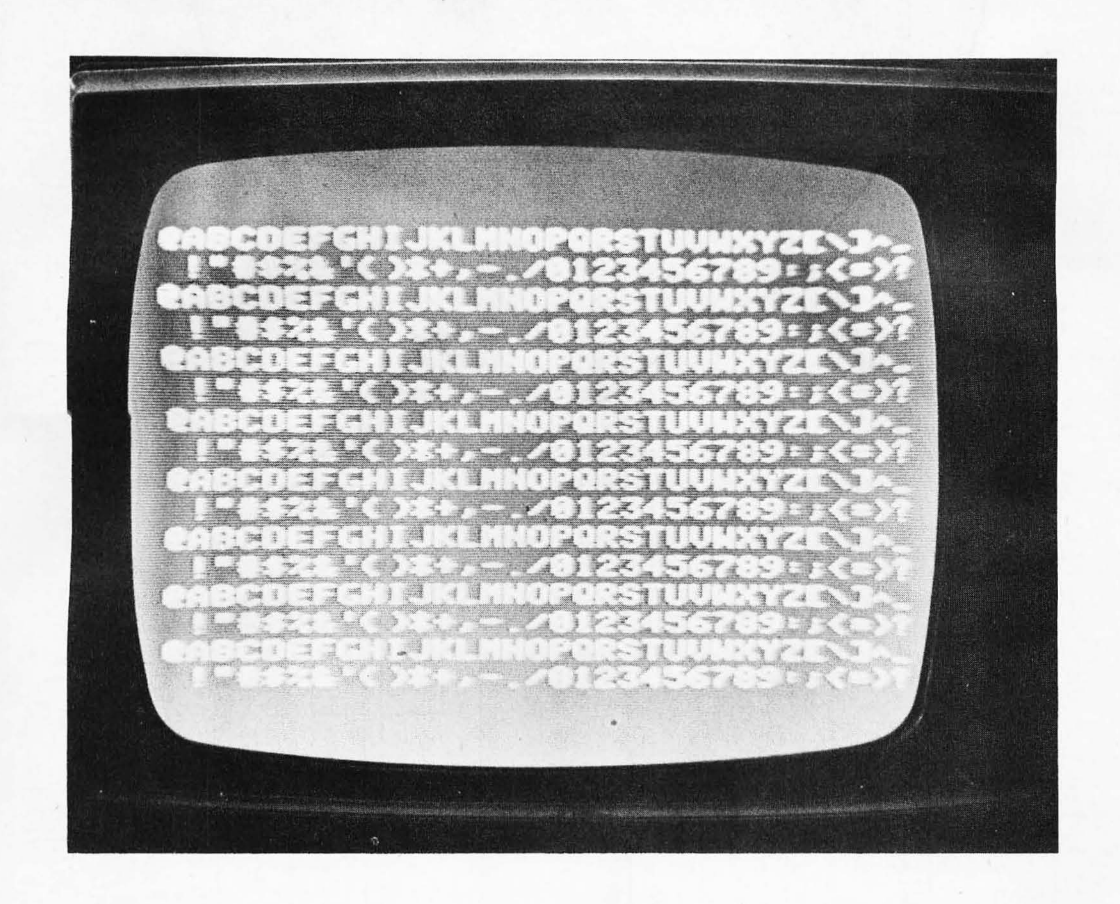


Photo 1: This is a test display pattern generated by connecting the low order outputs of the character and line counters to the character generator ROM's 6 input bits. The result presents every combination of the character set, so every character pattern is visible on the screen.

controls the brightness of the beam as it is swept across the screen.

To trace out a frame, the electron beam is slowly deflected from the top of the screen to the bottom as it rapidly sweeps horizontal lines. This vertical sweep is allotted 16.67 milliseconds (60 Hz) so there are 262½ lines in one frame. In a manner similar to the horizontal sync, the vertical sync causes the beam to be returned to the top of the screen to start a new frame. The beam is blanked during vertical retrace which takes about 1250 μs. This leaves 242 usable lines in each frame.

A complete picture is formed by two consecutive frames that are interlaced with each other. Interlacing means that the horizontal lines of one frame fit in between the horizontal lines of the other frame. The result is 30 complete pictures every second of about 484 usable (525 total) lines each. Because of the interlacing, however, the screen is illuminated at a 60 Hz rate. This eliminates an objectionable "flicker" that would be seen if the screen were only scanned at a 30 Hz rate.

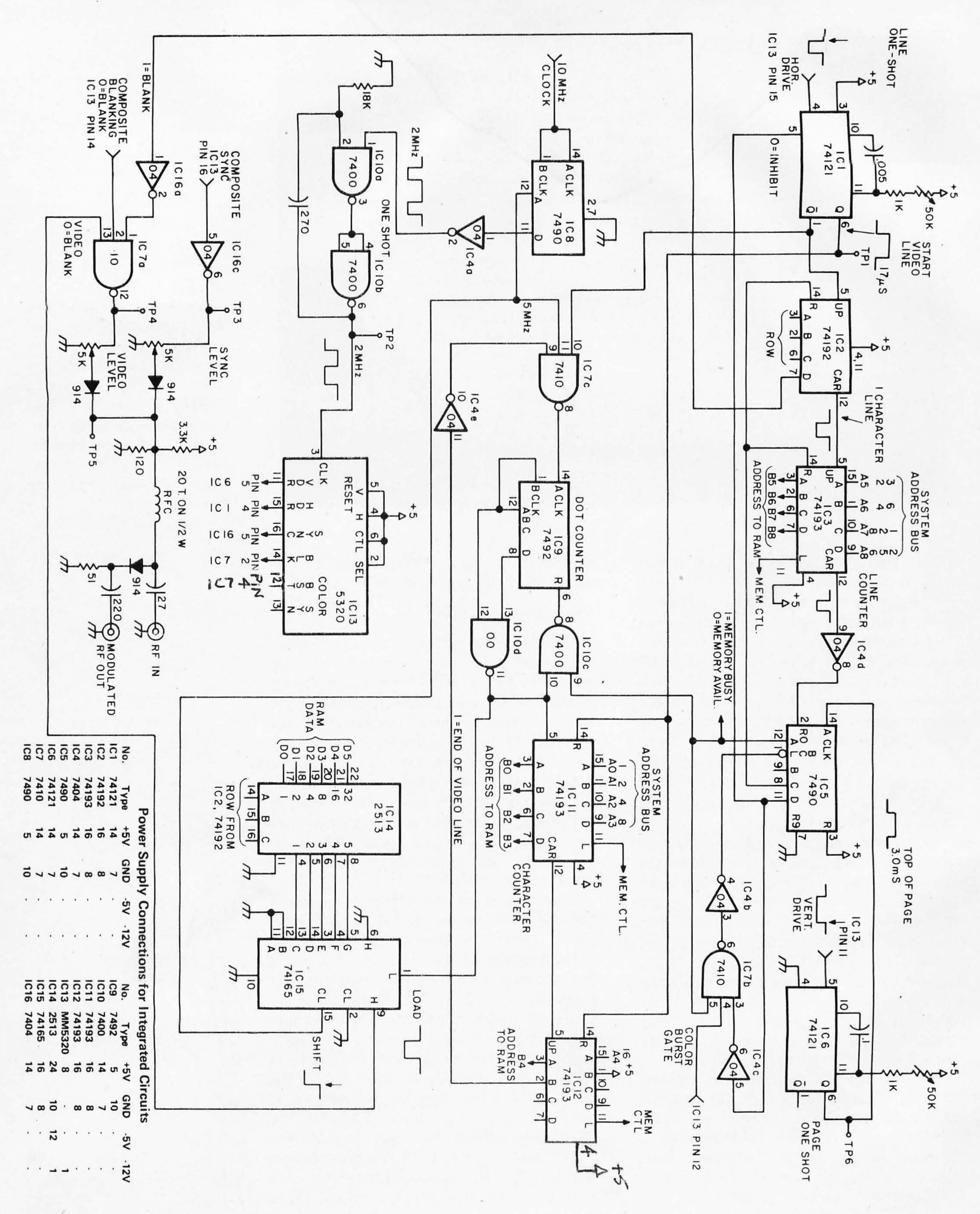
The TV signal received at the antenna terminals contains the information needed to generate the vertical and horizontal sync, blanking, and video. The TVD simulates a TV signal by supplying a composite wave form containing the same information normally present except sound. The full schematic of this TVD design (except memory) is shown in figure 2.

Character Generation

The scanning nature of the TV raster requires that the video (or brightness) information be sent in serial form to control the electron beam as it sweeps lines across the screen. Suppose, for example, that the character "H" is to be displayed as shown in figure 3. The first line can be represented as 10001, ones signifying light spots (dots) and zeros signifying dark spots. The remaining six lines can similarly be represented as a series of dots and dark spaces. When the seven lines are displayed one above the other, the character "H" is seen.

The tedious job of deciding where to put the dots (ones versus zeros) to generate a given character is done by the 2513 read only memory, IC14. It has been mask programmed at the factory with the bit patterns required for 64 separate five by seven dot matrix characters. The 2513 supplies five bits of parallel output data representing one line of a given character. It requires the six bit ASCII subset code of the character and the three bit line number as inputs. The five bit parallel output of the 2513 is converted to serial data by the 74165 shift register, IC15. To produce one line of video, five bits are required for each character in the line, plus spacing bits. Note

Figure 2: Schematic Diagram of the TV Display. This diagram includes details of the time base generation circuitry and control logic for television display generation. It omits detailed wiring of the memory circuits shown conceptually in figure 4.



Power Supply Connections for Integrated Circuits

No.	Type	+5V	GND	-5V	-12V
IC1	74121	14	7	.	.
IC2	74192	16	8	.	.
IC3	74193	16	8	.	.
IC4	7404	14	7	.	.
IC5	7490	5	10	.	.
IC6	74121	14	7	.	.
IC7	7410	14	7	.	.
IC8	7490	5	10	.	.
IC9	7492	5	10	.	.
IC10	7400	14	7	.	.
IC11	74193	16	8	.	.
IC12	74193	16	8	.	.
IC13	MM5320	8	.	.	1
IC14	2513	24	10	12	1
IC15	74165	16	8	.	.
IC16	7404	14	7	.	.

that the TVD generates two identical interlaced frames to make a complete picture. The result is that each character is actually 14 lines high.

Sync Generator

The MM5320 sync generator chip, IC13, uses a single 2.0 MHz input to produce all the sync and blanking signals needed for a 525 line interlaced raster. The same logic could be wired using TTL but would require considerably more hardware and probably cost just as much. (The 5320 runs $4.95 ppd from NEXUS Trading Co, Box 3357, San Leandro CA 94578.) The only disadvantage found thus far with the 5320 is that it prefers a square wave 2 MHz source. To this end the 100 nanoseconds pulse from the 7490 "D" output is squared using two 7400 sections of IC10 as a oneshot.

Line Generation

Horizontal drive (coincident with horizontal sync) from the 5320 triggers a 74121 oneshot, IC1, to delay the start of each line and establish the left hand margin on the screen. The output of the oneshot serves three purposes:

1. Triggers the 74192 row counter, IC2.
2. Resets the 74193 character counters, IC11 and IC12.
3. Inhibits the dot counter, IC9, until the start of the line.

When the line oneshot output pulse ends, the dot counter starts counting at 5 MHz. It resets itself every seventh count to allow for the five dots of the character plus a two dot space between characters. When the dot counter resets, it also loads the next character into the 74165 shift register, IC15. (The very first character of each line is all zeros since the 74165 is not loaded until the dot counter resets the first time.) The 74165 shifts out the two dot space and the five dot character at a 5 MHz rate. As each character is loaded, the 74193 character counter increments by one to change the address for the RAM to the next character. When the 32nd load pulse occurs, the 5 MHz input to the dot counter is inhibited using the "B" output of the second 74193 character counter. The 74165 continues to shift out the 32nd (last) character and then shifts out a steady zero. When the character counters are reset at the start of the next line, the process repeats itself.

Line Counter

The 74192, IC2, counts each video line displayed. It counts to 10 for the seven lines of character information plus a three line space.

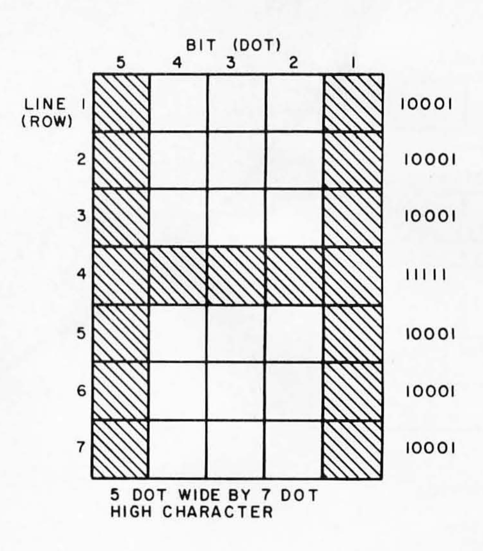


Figure 3: An example of a dot matrix pattern generated by the television display.

The "A", "B", and "C" outputs control the row inputs to the 2513 character generator chip. The first video line is all zeros since the row input to the 2513 is zero. Lines 9 and 10 are blanked using output "D" of the 74192, resulting in a total of three lines blanked. At the end of each complete line of characters, the 74193 line counter, IC3, increments by one until, at the end of the 16th line, a carry pulse is produced. This carry pulse resets the 7490, IC5, and signifies the end of a page. Output "A" of the 7490 is used to inhibit the 7492 dot counter and prevent the first line from being repeated at the bottom of the page.

Page Control

The 7490, IC5, stays reset until the top of the next page. Output "A" can be used to tell the memory control circuits that the TVD is not using the memory so that any required updates may be made by the CPU. Output "A" also inhibits the "B Clock" input via the 7410. The "D" output inhibits the line oneshot.

When a vertical drive pulse (coincident with vertical sync) triggers the 74121 page oneshot, IC6, the TV set syncs to the top of the next frame. The page oneshot delays the start of the first line to establish the top margin. At the end of the oneshot's output, IC5, bit "A" is clocked to a one. This tells the memory control that the TVD needs to resume control of the memory address and also enables the IC5 "B Clock" input via the 7410, IC7. The "B" section of the IC5 then proceeds to count color burst gate pulses to give the memory time to complete any access already in progress. The color burst gate was used only because it was convenient and occurs at the same rate as the horizontal drive – the horizontal drive could be used at the expense of a buffer since the 5320 can

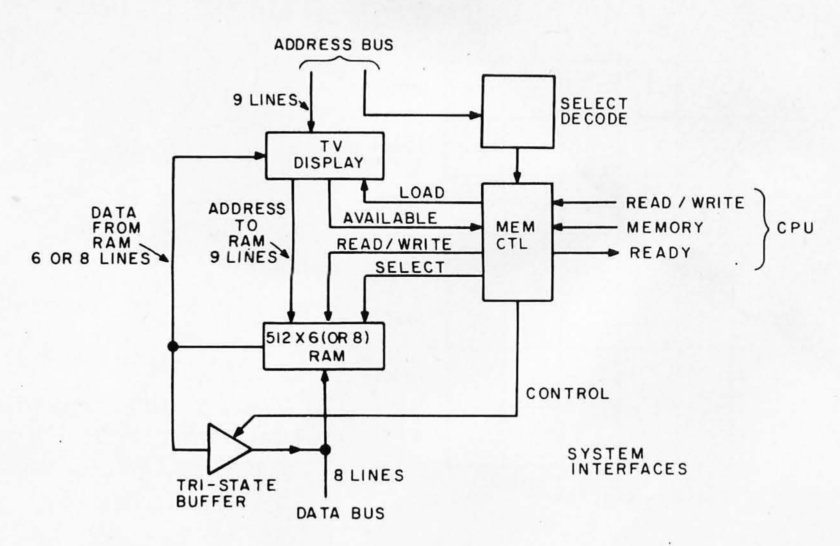


Figure 4: System Diagram. This figure details how the TV display fits into a central processor's memory address space. The low order 9 lines of address go directly to the line and character counters of the TVD; the memory array is addressed by the outputs of the counters, which are connected logically to the address bus when the load line demands central processor access. The high order bits of the processor's address are decoded separately and are used to enable processor access if the TV display portion of address space is referenced.

only drive one TTL load per output. When output "C" of the IC5 goes high, the line counters are reset. When output "D" goes high the "B clock" is inhibited via IC7b, and the line oneshot is enabled. This allows the first line to start.

Composite Video Generation

The video and sync are independently adjusted and then added to produce composite video. This can be piped directly into a set (be sure not to touch a hot chassis!) or used to modulate a low power RF source. A signal generator works fine for tests. (See "Television Interface" by Don Lancaster, page 20, October 1975 BYTE, for a thorough discussion of the various tricks to improve the interface.)

Memory Interface

Figure 4 illustrates how the TVD fits into a larger system. It is intended that the address outputs of the 74193 character and line counters (IC3 and IC11) be hard wired to the address lines of a 512 or 1 K by 6 static random access memory using 2102s or similar parts. The data outputs of low order 6 bits of this memory are the ASCII character select inputs to the 2513 character generator, IC14, and can be gated back to your system's data bus if you want the CPU to be able to read from the RAM. (Of course, a 512 by 8 memory would be needed if the CPU is to be able to use the RAM for other tasks.) The data inputs of the RAM tie to the data bus to allow the CPU to write into the RAM.

To avoid breaking up the picture on the display during access, the memory control logic must use the "A" output of IC5 to tell when the CPU can use the RAM and when it must signal a busy to the CPU at the start of a page. There is more than ample time between the "A" output and the line counter reset to finish any access in progress. To use this feature, the memory busy line must be wired to your processor's "memory ready" line (possibly through an inverter if the logic of your particular computer requires it). This method will work well for any processor, like the 8008 or 8080, which allows unlimited "memory busy" delays. However, for dynamic processors such as the 6800, the maximum processor delay time of about 5 μs dictates use of an alternative approach. One simple approach is to ignore the effect of memory access on the display. The result will be a short glitch in the display corresponding to each computer access. The nature of the glitch will be a resetting of the line and character counters to a new location, causing a scrambling of the display for the remainder of the current frame. A second approach is to wire the memory ready line into a single bit input port which can be tested as a status flag: If the line indicates a retrace, then the memory access software for the display will allow an update to occur.

The CPU addresses the RAM through the character and line counters (IC3 and IC11) by tying their data inputs to the system address bus and using the load control of pin 11. The 74193s can also be used as temporary storage for the address in a system with a common address and data bus. Note that the TVD does not interfere with CPU access to the remainder of the system's memory at any time and only delays the CPU by one of the techniques discussed above if it tries to access the TVD RAM while a page is being displayed. The CPU has the entire vertical retrace to make updates at once every 16.67 milliseconds.

Lacking a memory for my initial testing, the 2513 data inputs were temporarily tied to the 74193 address outputs (2513 PIN 17 to character counter PIN 3, etc) to display the complete 2513 repertoire every two lines as in photo 1. The 74193 load lines must also be connected to a "one."

Modifications and Adjustments

1. There is one known bug so far and no doubt more will show up when the TVD is integrated into a system. The 7490 can, on power up, hang in state with both the "C"

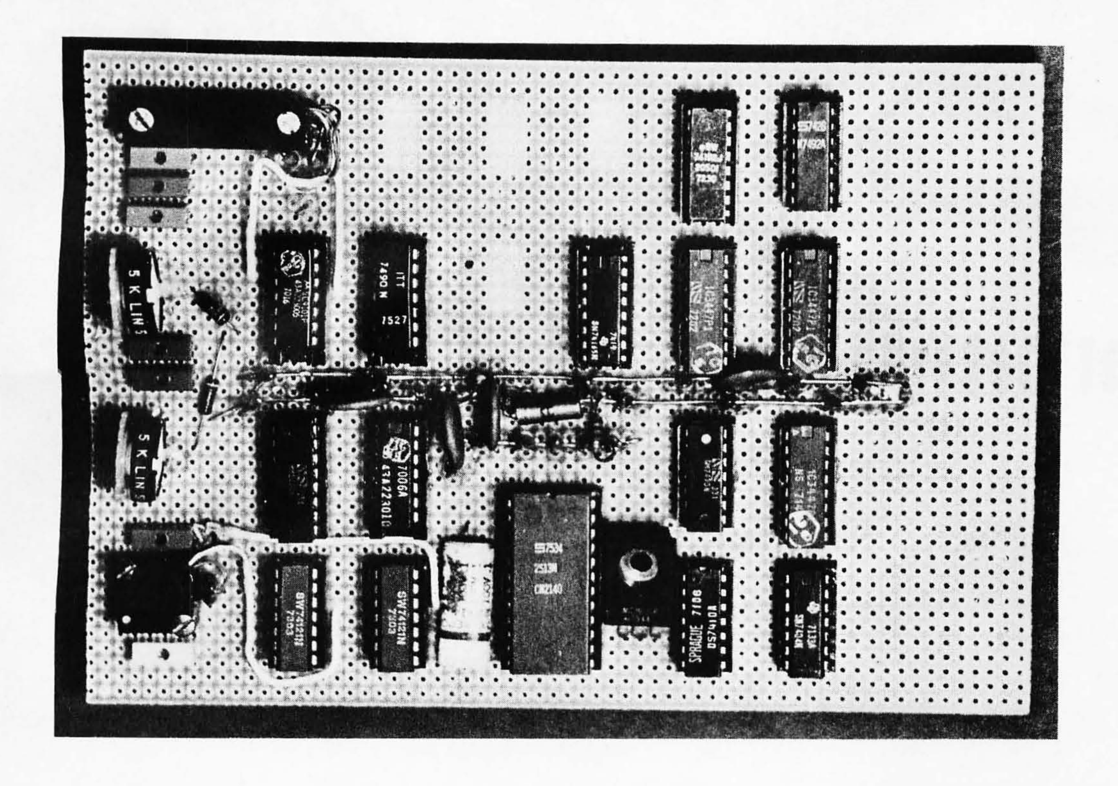


Photo 2: Prototype Circuit. The large socket is for the character generator. Test points are the 6 small rectangular objects along the left hand side of the board. A 7812 regulator in a plastic package is to the right of the character generator, and is used to provide the –12 V bias for the ROM. A zener diode with a dropping resistor is used to create the –5 V bias required for the ROM.

and "D" outputs a one. This state continuously resets the 74192 row counter and is nonrecoverable. A cure would be to power-on-reset the 7490 "R9" which does produce a recoverable state.

2. As mentioned before, the 5320 likes a square wave input, so check the 7400 imitation of IC10a and b oneshot, or better yet use a 74121.

3. Using a separate oscillator for character generation (5 MHz) would allow adjustable character width, but watch out for any interaction with the 2 MHz – it shows up as a torn, garbled display, as will most sync or jitter problems. A crystal is best. (It is possible to use 12 MHz and a 7492 in place of IC8 to get 6 MHz for the characters and 2 MHz for the 5320.)

4. The prototype is wire wrapped on a 4½ by 6 inch (11.43 by 15.24 cm) vector board (see photo 2) with room to spare, although a slightly larger board would accommodate more interface goofs. Fulp's corollary says things like this always get bigger. Also, the 44 pin connector planned for the prototype is not large enough counting the additional RAM address and data lines.

5. The modulation levels for the radio frequency modulator are fairly critical and misadjustment of sync or video levels will cause a torn display. Try setting video level control for 1/2 of maximum and sync for 3/4 of maximum.

6. Harmonics from the 10 MHz tend to leak into the TV so pick a higher channel (5 or 6) if herringbone is noticeable on your display.

7. There are many causes for ghosting and smeared characters including VSWR (voltage standing wave ratio) on the cable to the TV, misadjusted fine tuning, or a narrow band width TV.

8. Character and line spacing can be altered by modifying the dot and row counters, respectively, to reset at different counts. Be careful though, or the display will not fit on the screen.

9. The unused bits C and D of the second 74193 character counter, IC12, may be used for the 512 and 1024 bits if a two or four page RAM is desired. Some method of controlling these bits during display time is needed to select the page.

10. A light pen could strobe the present RAM address into a latch to be read by the CPU via the data bus.

11. The 5320 provides color sync gating so how about color characters? The extra 2 bits available with an 8 bit wide RAM can provide software control of many goodies (like brightness, color, blinking, underlining, black on white, etc).

The TVD as described can be used to display one or more pages of ASCII characters and opens up many possibilities of modernizing the IO portion of a small system. My home brew computer will be a complete microcomputer chip based system, designed with the TVD as the main man machine interface.■

Programming for the Beginner

A Structured Start

Ronald T Herman
Simpson Rd
RFD 1 Box 125
Windham NH 03087

A program can be viewed as an edifice built from the bricks of SEQUENCE blocks, and the mortar of IFTHENELSE, DOWHILE, DOUNTIL and SELECT blocks.

For a number of years now the field of computer programming has been moving from the realm of a black art to an organized and systematic process. A number of programming techniques have evolved during this change. This article will present the basics of a technique known as structured, top down programming. In the process of applying these techniques in my own work, it occurred to me that the basic concepts could be useful to those just learning to program, not to mention the veteran hackers in the crowd. If learned at an early stage, these techniques can lead to more rapid and sound development of one's programming skills.

A structured approach to program development has among its virtues the following points:

- It allows the novice programmer to get acquainted with programming logic without having to be concerned with a specific machine or programming language. It allows him to grasp the flow of a program without worrying about bits and bytes.
- Followed correctly, structuring can lead to a program that is relatively free from logical errors the first time it is coded and relatively easy to debug once it is run on the machine.
- Pseudo code, a byproduct of structuring, allows a means of exchanging program ideas with others, regardless of the machine with which they might be familiar.
- Pseudo code provides a convenient alternative to flow charts that can be incorporated into a program listing as comments for future reference and explanation.

This process of getting things done in an organized fashion has its drawbacks. However, most of these seem to be psychological. Properly applied structured technology tends to minimize one of the facets of programming that has attracted many in the past: the chance to see how cleverly and concisely one can write a software routine. This seems to have been replaced by the challenge of trying to write a routine in a straightforward manner and at the same time trying to rigidly follow a set of fairly simple rules.

What will be presented in this article are some of the basic building blocks of structured programming and an example illustrating the design of a simple program using these blocks.

The Building Blocks of Structure

So much for the sales pitch. What then is structuring? Some number of years ago it was shown that a program could be built from a set of simple building blocks all having the property of one input and one output. While not everyone agrees on what composes this set of building blocks, the one in, one out property is common to all. Presented here are a few of the most common examples that should cover most situations.

The SEQUENCE Block

Probably the simplest (and most trivial) unit of structure is the SEQUENCE. This is illustrated in figure 1 and is nothing more than one process performed after another.

The IFTHENELSE Block

One of the powers of a computing machine is to make a decision based on a set of conditions and take a specific action as a result of that decision. This capability is represented as the IFTHENELSE block shown in figure 2. In the figure, "p" is an expression or some set of conditions. In a checking account, for example, one adds deposits and subtracts checks written. An IFTHENELSE statement of this fact would appear as follows:

```
IF (transaction is a deposit) THEN
:   (add amount of transaction to balance)
ELSE (subtract amount of transaction from balance)
ENDIF
```

Here is our first example of writing a program step in a machine independent "pseudo code." The format of pseudo code is mostly a matter of taste. The punctuation is optional, but the indentation is necessary for readability where many complex IFTHENELSE decisions are grouped together. Some people use asterisks (*) instead of colons (:) to mark margins and some omit the parentheses around descriptive phrases. The ENDIF helps clarify the limit of operations within a more complex statement. Each statement line represents a process to be performed or a condition to be tested. The statement or condition preferably should not be continued on another line.

The DOWHILE Block

The decision making capability of computers, combined with the ability to change the order in which instructions are executed, provides an even more powerful feature – the ability to repeat a calculation or series of operations many times. This capability is represented in the DOWHILE building block shown in figure 3. The DOWHILE is just a special application of the IFTHENELSE given earlier. In a DOWHILE block, a process is done as long as a set of conditions "p" is true. Note that the condition is tested first before the process is performed. Suppose you have 10 transactions to update into your checking account, some checks written and some deposits. In pseudo code this becomes:

```
(set counter to number of transactions)
DO WHILE (count is non zero)
:   (process the transaction)
:   (decrement the count)
ENDDO
```

Note that the DOWHILE is terminated by an ENDDO. The "(process transaction)" statement could be the IFTHENELSE given above. If combined, the result would be as follows:

```
(set counter to number of transactions)
DO WHILE (count is non zero)
:   IF (transaction is a deposit) THEN
:   :   (add amount of transaction to balance)
:   ELSE (subtract amount of transaction from balance)
:   ENDIF
:   (decrement the count)
ENDDO
```

The DOUNTIL Block

The DOUNTIL block is shown in figure 4. It differs from the DOWHILE only because the condition "p" is tested after the process is performed. This can simplify the writing of machine code from pseudo code. Suppose one wanted to read characters from a keyboard until a carriage return is encountered. It could be done with a DOWHILE by saving the last character read as follows:

```
(clear last character read)
DO WHILE (last character not a carriage return)
:   (get a character from the keyboard)
:   (save character in last character read)
ENDDO
```

Using structured programming concepts, many logical errors and bugs can be caught at an early stage in the design process.

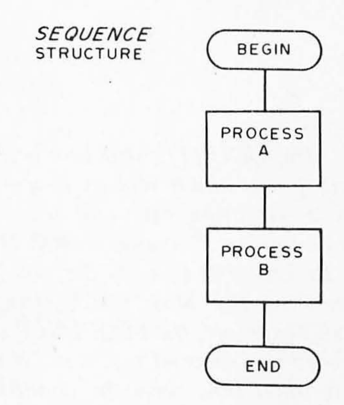


Figure 1: The SEQUENCE structure is a series of self contained processing steps which are executed one after another. Flow in this diagram begins at the top and proceeds down the diagram. The number of steps defined in a SEQUENCE block is arbitrary; the example here shows two steps, A and B. In this article's figures, the notation BEGIN and END is used to mark the well defined extrance and exit points of the structures depicted. (NOTE: Processes A and B may be more complex combinations of the building blocks in all of these figures.)

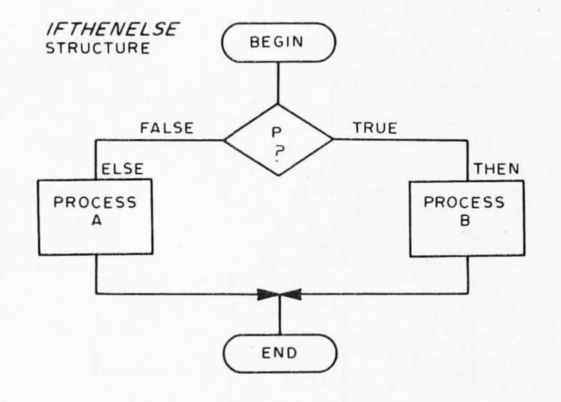


Figure 2: The IFTHENELSE structure is a conditional test and two alternative SEQUENCE structures. The THEN alternative is executed if the condition, P, is found to be true. In this illustration, the THEN alternative is shown as a one step SEQUENCE structure called B. The ELSE alternative is executed if the condition is found to be false. In this illustration, the ELSE alternative is shown as a one step SEQUENCE structure called A.

DOWHILE
STRUCTURE

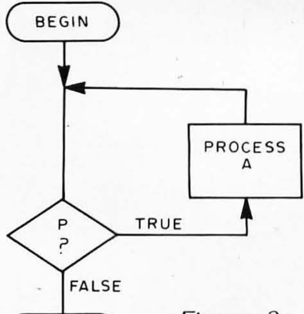


Figure 3: The DOWHILE structure is a looping form which repeats a specified SEQUENCE structure over and over again as long as a condition, P, is true. DOWHILE tests the condition prior to executing the SEQUENCE structure for the first time. Thus in this example, the SEQUENCE structure A could be executed 0, 1, 2 . . . N times, depending upon how soon the condition P becomes false as a result of A's work.

DOUNTIL
STRUCTURE

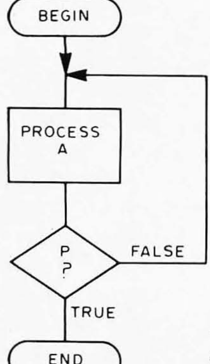
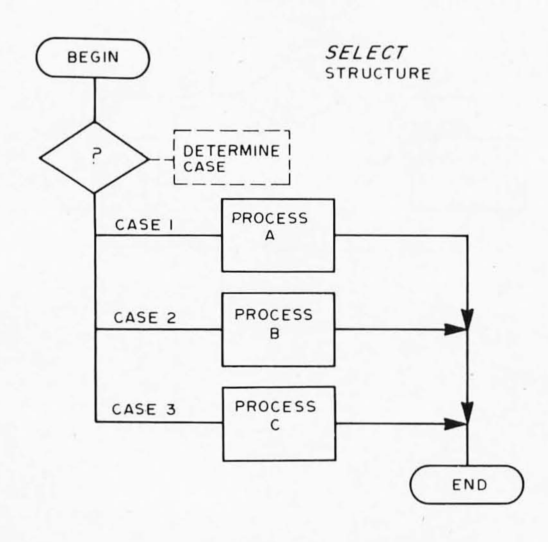


Figure 4: The DOUNTIL structure is another looping form which repeats a specified SEQUENCE structure over and over again until the condition, P, is true. DOUNTIL, in contrast to DOWHILE, tests the condition after executing the SEQUENCE structure. Thus in this example, the SEQUENCE structure A could be executed 1, 2, 3 . . . N times depending upon how soon the condition P becomes true as a result of A's work.

Figure 5: The SELECT structure is a more comprehensive version of the IFTHENELSE concept; it allows data to be tested for multiple cases. The result is the picking of one of "N" cases. In this example, N is 3, so there are three SEQUENCE structures which might be executed depending upon the case determination.



This would require an extra instruction or two when translated into machine code, since the "last character read" must first be initialized to contain *something* other than a carriage return. Implemented as a DOUNTIL it is simply:

```
DO UNTIL (character read is a carriage return)
:    (get a character from the keyboard)
ENDDO
```

The SELECT Block

Sometimes it is necessary to select one of many possible processes based on some quantity that may take on any number of values. Suppose, in addition to updating your checking account balance, you decided to keep a tally of money spent on each of several budget items such as food, medical, car, electric and so forth. This could be done with a string of IFTHENELSEs as follows on the next page. Two possible methods are shown but both are somewhat awkward to follow.

```
IF (check was written to super market) THEN
:    (add amount to food total)
ELSE
:    IF (check was written to doctor) THEN
:    :    (add amount to medical total)
:    ELSE
:    :    IF (check written to auto repair shop) THEN
:    :    :    (add amount to car total)
:    :    ELSE
:    :    :    IF (check written to electric company) THEN
:    :    :    :    (add amount to electric total)
:    :    :    ENDIF
:    :    ENDIF
:    ENDIF
ENDIF
```

Alternate method:

```
IF (check written to super market) THEN
:    (add amount to food total)
ENDIF
IF (check written to doctor) THEN
:    (add amount to medical total)
ENDIF
IF (check written to auto repair shop) THEN
:    (add amount to car total)
ENDIF
IF (check written to electric company) THEN
:    (add amount to electric total)
ENDIF
```

A more concise and meaningful way to describe this process is with the SELECT block shown in figure 5. Note that although there are many paths through the block, there is only one entrance and only one exit. Our bookkeeping example now becomes:

```
SELECT (based on who check written to)
:    CASE (written to super market)
:    :    (add amount to food total)
:    CASE (written to doctor)
:    :    (add amount to medical total)
:    CASE (written to auto repair shop)
:    :    (add amount to car total)
:    CASE (written to electric company)
:    :    (add amount to electric total)
ENDSELECT
```

These then are the building blocks of a structured program. Others could be invented, but these should suffice for most situations. In any case, each should exhibit one entry point and one exit point. It should be noted that none of the building blocks

transfer control (jump) into another, never to return. This so called GOTO is a definite "no no" in structured programming. All processes are either done in line or are called as subroutines that are presented elsewhere. Frequent jumping around in a program results in a maze of paths that becomes difficult to follow and even more difficult to deal with in the event that a change in one is necessary.

Building From the Top Down

Earlier when the subject of structure was introduced, the term "top down" was used. If you wanted to build a computer, you could start by getting the processor, then some memory and IO devices and a power supply. Then you would have to try to determine how to connect all the parts together. On the other hand, you could start by deciding what the specifications for the machine are to be, such as word length and speed, what the IO ports look like and what controls and devices are to be attached. From there the problem is to select or design the components and parts to do the job.

So it is with software. In the past the tendency has been to first develop the pieces like Teletype handlers, tape read/write subroutines and others. Then the pieces would be fitted together into a functioning module, hopefully without having to make any major changes to the pieces already developed. The experience of many people in the professional software field has indicated that this is not an efficient way to design a software module. Instead the approach is to start at a high level of abstraction to describe the basic function to be performed. From there each unit of this description is broken into more detailed modules. Once designed, the program is coded and debugged a piece at a time starting at the topmost level. Subordinate levels of code are temporarily replaced by dummy "stubs" which do nothing. Then as each level is coded and incorporated into the program, any problems that develop usually can be isolated to the modules just added.

As an example of this approach and the use of pseudo code, let us design a simple editor program. This editor reads a line of text from an input device (paper tape reader or magnetic tape recorder). The line is saved in memory and displayed on a video monitor or typed on a Teletype printer. A limited number of responses from the input keyboard allow changes, deletions, and insertions to be made. Upon completion, the line is written to the output device (punch or another magnetic tape recorder). The process continues until the end of tape is reached on the input device. Changes and insertions are made by typing the character on the Teletype directly below the input line. Inserts are indicated by terminating the line with a carriage return (CR) and changes by a line feed (LF). The Teletype carriage or video display cursor is positioned using a "Control P" character (holding the CONTROL key down while striking the "P" key). This is not a sophisticated editor, but should serve as a good example of how to use the techniques described.

The topmost abstraction level of the editor program can be described in pseudo code as follows:

```
DO UNTIL (end of input tape)
:   (get line from input and type on printer)
:   (get response line from keyboard, store and echo it)
:   IF (only CR or LF entered) THEN
:   :   (do nothing)
:   ELSE
:   :   IF (last character is LF) THEN
:   :   :   (do character changes and output line)
:   :   ELSE (do character inserts and output line)
:   :   ENDIF
:   ENDIF
ENDDO
```

This then is our editor in its most abstract form. Note that an input line is deleted by entering only a carriage return or line feed. Now let us refine the description by describing each process identified above.

Getting a line from the input device requires turning on the input device, reading characters, and storing them until a line feed or carriage return has been recognized. The stored line is terminated with a zero (null) character so that the end of the line is more easily recognized later.

```
(set input line pointer to first address of line)
(turn on input device)
DO UNTIL (a LF or CR is read)
:   (get character from device)
:   (store character @ input line pointer)
:   (advance input line pointer one position)
:   (send character to printer)
ENDDO
(clear a character at the pointer address)
(turn off input device)
```

Likewise getting the response from the keyboard is similar except that Control P characters are echoed as spaces on the Teletype printer.

```
(set keyboard line pointer to first address of line)
DO UNTIL (LF or CR is typed)
:   (get character from keyboard)
:   IF (character is not a LF or CR) THEN
:   :   (store character @ keyboard line pointer)
:   :   (advance keyboard line pointer)
:   :   IF (character is not Control P) THEN
:   :   :   (echo the character on printer)
:   :   ELSE (echo a space)
:   :   ENDIF
:   ENDIF
ENDDO
(clear a byte @ keyboard line pointer)
```

Character replaces and inserts are done by using the Control P characters on the keyboard to indicate where the changes are to be made. For each Control P character in the response, an input line character is sent to the output. When a character other than Control P is encountered, it is either inserted into the output or replaces a character about

For a number of years, the field of computer programming has been moving from the realm of a black art to an organized and systematic process.

"Top down structured programming" is a veritable buzzword in the data processing and computer science fields.

to be outputted depending on the last character from the keyboard (line feed or carriage return). Thus the replace operation becomes:

```
(set input line pointer to start of input line)
(set keyboard line pointer to start of keyboard line)
(turn on output device)
DO UNTIL (end of keyboard line)
:   (get keyboard line character @ keyboard line pointer)
:   IF (character is Control P) THEN
:   :   (get character @ input line pointer and send to output)
:   :   (echo character on teletype printer)
:   ELSE (send the keyboard character to the output)
:   :   (echo the keyboard character on printer)
:   ENDIF
:   (advance keyboard line pointer)
:   (advance input line pointer)
ENDDO
(put out rest of characters in input line)
(turn off output device)
```

Structured programming is a systematic way of thinking about processes, the result of which is a well designed and understandable program specification.

Note that the resulting output is echoed on the Teletype to enable verification of the operation.

The insert operation is given below:

```
(set input line pointer to start of input line)
(set keyboard line pointer to start of keyboard line)
(turn on output device)
DO UNTIL (end of keyboard line)
:   (get keyboard character @ keyboard line pointer)
:   IF (character is a Control P) THEN
:   :   (transfer character @ input line pointer to output)
:   :   (echo character on teletype printer)
:   ELSE
:   :   DOWHILE (keyboard character is not Control P)
:   :   :   (send keyboard character to output)
:   :   :   (echo keyboard character on printer)
:   :   :   (advance keyboard line pointer)
:   :   ENDDO
:   :   IF (NOT END OF KEYBOARD LINE) THEN
:   :   : (transfer character @ input line pointer to output)
:   :   : (echo character on teletype printer)
:   :   ENDIF
:   ENDIF
ENDDO
(put out rest of input line characters)
(turn off output device)
```

The routine that "puts out the rest of the input line characters" is:

```
DO UNTIL (input line pointer points to a null)
:   (get character @ input line pointer)
:   IF (character is not a null) THEN
:   :   (put character to output device)
:   :   (echo character on printer)
:   :   (advance input line pointer)
:   ENDIF
ENDDO
```

This article concerns organizing and planning a program, which is expressed in a structured "pseudo code." The next step after the plan is created is to translate the pseudo code into the detailed machine code of your personal computer.

Finally the routines to get a character from the input device and keyboard in this simple system are identical except for the address of the device referenced.

```
DO UNTIL (input device ready flag is on)
:   (get input device ready flag)
ENDDO
(get character from device data port)
```

The character output and type routines are likewise the same.

```
DO UNTIL (output device ready flag is on)
:   (get output device ready flag)
ENDDO
(send character to output device data port)
```

We have now arrived at such a level of detail that the code could be written without much difficulty from the pseudo code on an almost one for one basis. Each module except for the top level description could and probably would be written as a separate subroutine. Note that each module can be read starting on the first line and ending on the last. No transfers are made out of any module to another without returning to the line following. Modules should be kept short (no more than a page) so that they can be read without constantly flipping pages back and forth.

Conclusion

What has been presented in this article is a description of a systematic approach to program design and a means of describing it so that almost any individual should be able to understand it. The resulting program when coded will have been well thought out and may even have been reviewed and partially debugged by other individuals not intimately familiar with the machine upon which it will ultimately be executed.

Much discussion has occurred about standards for data exchange between various computer hobbyists. On a higher level, the pseudo code approach makes possible a standard way to exchange program ideas. In fact, higher level languages have been developed that, at least in part, resemble the pseudo code language used here. Using this approach, programs might be written to convert pseudo code into machine instructions for the 8080, 6800, 6502 or other CPUs as they become available. All hobbyists could then share programs in a higher level language, each doing the necessary conversion on his own machine.

There are a number of references on the subject of structured programming. The idea has been discussed extensively in computer science circles in recent years, to the point that "structured programming" has become a buzz word in the business. This writer is familiar with the two texts given in the bibliography. The IBM text is excellent for beginners and those new to the concepts, while the McGowan and Kelly text is a more rigorous and mathematical presentation.■

BIBLIOGRAPHY

International Business Machines Corp, *Structured Programming Independent Study Program*, Poughkeepsie NY, 1974.

McGowan, Clement L and Kelly, John R, *Top Down Structured Programming Techniques*, Petrocelli/Charter, New York, 1975.

Why haven't YOU ordered YOUR SPHERE COMPUTER yet?

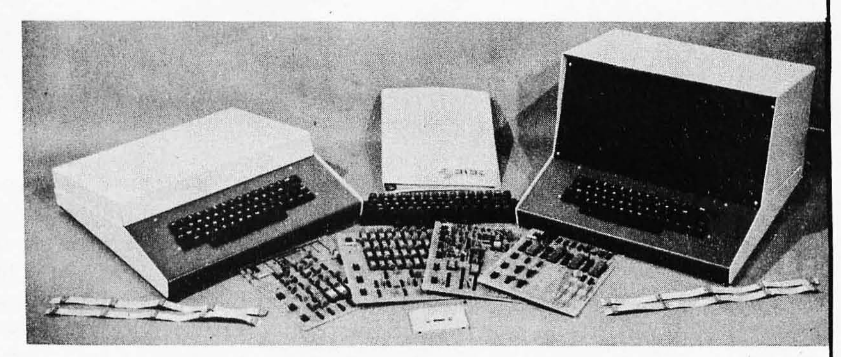
....Probably because you haven't understood what a SPHERE COMPUTER can do for you, and what you get for the money you spend. One of the most important things to consider when spending your hard-earned money is how much value you will receive out of each dollar you spend. SPHERE CORPORATION offers you a COMPLETE COMPUTER SYSTEM. From one vendor you can get all the interfaces and peripherals that allow you to perform similar functions of big computers. Remember, you spend a little more, but you can do so much more than you can by saving a "buck". What will a SPHERE COMPUTER do for YOU?

TRADITIONALLY, a computer must allow you to input some information or problem into the machine for processing, then produce a solution. WHY USE A MACHINE?' ...Because a computer can do it faster, more efficiently, more accurately and with a lot less "fuss" than a pencil or culculator. A computer can be programmed to make an analysis of the situation and produce a solution after having considered the variables and contingent factors.

PROFESSIONAL, or NON-PROFESSIONAL, a SPHERE COMPUTER can do the job! Whether your needs be to save time processing the payroll records, to produce a daily inventory of goods on hand, or to regulate and monitor the operation of sensitive machinery, SPHERE can handle the job. How about preparing the monthly accounts receivable? ...Did you ever make a transposition error that cost you hundreds of dollars? SAVE TIME WITH A SPHERE COMPUTER! ....Remember, it doesn't take a coffee break, get sick, get pregnant or ever go on vacation! The computer does what you tell it to do. (Just remember to tell it the right things). Your SPHERE all-purpose, professional quality computer can be configured to your needs and will feel right at home in the office, school, or Lab.

WHAT ABOUT NON-PROFESSIONAL COMPUTER USERS? Well, we have found that these are people who don't rely on a MICROCOMPUTER to "Bring Home the Bacon". Whether your needs be to design a new method to help you out at work, or to file and reference a thousand valuable magazine articles, SPHERE has a model which will fit your needs, and that won't destroy your bank balance. Put all of your financial records into the computer and have it balance the check book and keep record of deductions you can use on your taxes. Get the computer to budget monthly expenditures and provide the best use of your investment dollars.

HEY, DID YOU EVER PLAY A COMPUTER GAME? ....Computer games make you THINK, which challenges and expands your mind. Chess, Roulette, Star Trek,...There are hundreds of science fiction fascinating games. The important thing is that a computer, when presented with your problem reviews the related variables and processes the data to give you results swiftly; and with a SPHERE SYSTEM, it is done in a manner which is easy to understand ...and USE! When you purchase your SPHERE COMPUTER SYSTEM you are receiving the highest quality machine of it's kind on the market.



KIT, OR ASSEMBLED, SPHERE IS A PROFESSIONAL COMPUTER

consider these features:

- EASY-TO-USE TYPEWRITER STYLE KEYBOARD
- DISPLAY OF ALPHA/NUMERIC CHARACTERS ON YOUR TV SET OR OUR VIDEO MONITOR.
- THE MOTOROLA 6800 MICROPROCESSOR WITH ONE OF THE FINEST INSTRUCTION SETS AVAILABLE.
- RANDOM ACCESS MEMORY STORAGE (4K to 64K BYTES)
- INTERFACES TO TTY (TELETYPE/PAPER TAPE) TTL, RS232, MODERN, LINE PRINTERS, FLOPPY DISK, AND STANDARD AUDIO CASSETTES.
- DOCUMENTATION TO TELL YOU WHAT TO DO.
- AND MANY, MANY MORE...(I'm out of room).

.....NOT ONLY DO YOU GET A WELL DESIGNED COMPUTER... but a unit which has your growth needs considered. As your needs change, and grow, so can a SPHERE COMPUTER! Expandability is no problem... thanks to sufficient BUSS CABLE CONNECTIONS. Simply add-on what modules you need. With a SPHERE COMPUTER product, your jobs will be accomplished in the most efficient manner possible. Don't worry about running around town looking for a peripheral, or having to depend on another company to get that extra device... (the one you didn't know you'd need) to make your computer work.

.....THE SPHERE PHILOSOPHY ...is to provide you with a complete product, at the start, and from one vendor. Input of data, problem solving and meaningful output; the hardware is complete.

SO, WHEN IT COMES TO MAKING YOUR DECISION....Remember that what you want is a computer which will solve your problems, save your money that's being wasted, and that most importantly...GETS THE JOB DONE!

WHY haven't YOU ordered YOUR SPHERE COMPUTER yet?

CONTACT SPHERE, or SEE YOUR LOCAL DISTRIBUTOR:

COMPUTER MART
473 Winter Street
Waltham, Mass.02154
617-890-0677

COMPUTER MART of N.Y.
314 5th Ave.
N.Y., N.Y. 10001
212-279-1048

COMPUTER WORKSHOP
11308 Hounds Way
Rockville, Md. 20852
301-468-0455

COMPUTER COUNTRY
18 Alameda Sq.
2200 W. Alameda Blvd.
Arvada, Colorado 80003
303-935-1100

THE SMALL BUSINESS COMPUTER COMPANY
400 Dewey Blvd.
San Francisco, Calif. 94116
415-665-2575

COMPUTER WAY
15525 Computer Lane
Huntington Bch, Calif.92649
714-892-8816

BARGAIN ELECTRONICS
2018 Lomita Blvd.
Lomita, Calif. 90717
213-539-2260

IVth DIMENSION, INC.
7060 Miramar Rd.
Suite 104
San Diego, Calif. 92126
714-566-7610

COMUNICACIONES S.A.
APDO 703
San Jose, Costa Rica
23-47-12

PMI
European Distribution
13, Chemin du LeVant
01210 Ferney-Voltaire
Ain, France
050-416222



SPHERE CORPORATION

940 North 400 East
North Salt Lake, Utah 84054
P.O. BOX 213 (801) 292-8466

NOTE: Some of the above applications and programs are available from Sphere.

A High School Computer System

Christopher Lett
Mac Gregor Dr
Mahopac NY 10541

We needed hard copy and a high level language . . . .

Homebuilt minicomputers such as the Altair 8800 offer an economical but efficient alternative to more expensive options.

In late May of 1975, John F Kennedy High School in Somers NY suddenly had to face a computational crisis. For the previous two years, the school had been given computing time gratis by the local Board of Cooperative Educational Services; our only expenses were the purchase of an acoustic coupler, and the telephone connection costs. But that May we were abruptly informed that the service would no longer be available.

The problem was that there were three courses that made use of the computer already scheduled for the following school year: an interdisciplinary course, a full year calculus course, and a course in BASIC language programming. Since it was too late to design new courses and drop these, the school began to search for an inexpensive computing system. Besides low cost, we needed a system with a powerful, high level conversational language (either BASIC or APL) with the ability to store programs in some form such as paper tape or audio cassette. The terminal had to provide hardcopy and come with a paper tape reader, if necessary.

Examining the Alternatives

The first possibility was purchasing computing time on a time sharing basis from a major corporation. This would have cost the school over $3000 per year, and was therefore rejected as being too expensive.

The second alternative was to purchase a self contained computing system, such as the IBM 5100 or the Wang 2200. Although they would have filled most of our requirements, their high initial costs ($9000 and $5400 respectively) made them again too expensive for our small private high school's tight budget.

The third and most probable choice was to buy a minicomputer with BASIC software and rent a teletypewriter to interface with it. Since a system of this type met our requirements at an absolute minimum cost, it was decided that this was the way to go. Now there was another important decision to make: What minicomputer system should the school purchase?

That summer, MITS Inc was running a sale on its Altair 8800 computer. What it offered was the Altair 8800 computer, two 4 K dynamic memory boards, an interface board, and, most importantly, their 8 K version of BASIC on paper tape, all for only $995. This meant that the system would pay for itself in less than a year, as compared with the next most expensive alternative. Table 1 shows the breakdown of costs we estimated during the summer of 1975.

The Teletype Model 33 ASR was selected for use as the terminal for several reasons: It provides hardcopy output, it has a paper tape punch and reader, it does not need a telephone connection; and we knew from previous experience that it is rugged and reliable, with maintenance, as needed, readily available under the leasing agreement.

Assembling the Altair

A check for $995 was subsequently mailed off to MITS in New Mexico, and we waited for the kit to come . . . and waited . . . and waited. After almost two months of patience, the kit arrived at the school in late October. Since my father and I were charged with actually building the thing, I had to bring the kit (data bus and all) home with me on the school bus (which was an experience in itself)!

The assembly manual for the Altair was somewhat disappointing in its handling of

Our high school suddenly faced a computational crisis: With computer courses all scheduled, we lost access to a "free" time sharing service.

I had to bring the kit (data bus and all) home with me on the school bus (which was an experience in itself)!

errata information. When MITS makes a change in one of the kits, it throws a pile of modification and errata sheets into the front of the manual. While the information is complete, this makes it hard to keep up with the changes that have been made. A better solution might have been to issue replacement "change pages" to be substituted for uncorrected originals. Another minor disappointment was the fact that not all the bugs had been caught by MITS. One such uncorrected mistake was the fact that the "+" and "–" signs on the power supply's bridge rectifier did not line up with the corresponding signs printed on the board itself. My father and I ended up having to trace the proper connections on the schematic to see what the correct alignment was. I believe that anyone who was unfamiliar with working from a schematic would have some trouble understanding how to orient that rectifier.

Other small problems included nuts, bolts, and screws that always seemed to be the wrong size for the job, and a shortage of terminal lugs.

Working nights and weekends, my father and I completed construction within two weeks. Powering up the kit for the first time, we discovered that the only defective part was one LED on the front display panel. The only thing left to assemble was the serial IO board. This time the assembly instructions were clear enough, but the theory of operation manual was somewhat sketchy.

One thing that MITS failed to mention was how to program the Altair to talk to a Teletype! You would think that they would mention that the interface must be set for 8 data bits, no parity bit, two stop bits, and device addresses 000 and 001, right? Wrong! This information was not mentioned in the documentation. Apparently MITS cannot tell you how to interface the Altair with any specific terminal because they have no way of knowing what kind of device you would be using in the first place. It is fortunate that we had read Don Lancaster's article on serial interfaces in the September 1975 BYTE. My

Table 1: Comparison of Two Year Computing Costs.

SYSTEM	TOTAL COST
IBM 5100	$9000
Wang 2200	$5400
Commercial timesharing	$6740
Phone line cost – $150 per month	
Computing costs – $100 per month	
Terminal with dialup – $87 per month	
Altair Package	$2195
MITS Altair plus software – $995	
Teletype Model 33 ASR – $60 per month	

recommendation on this point would be a set of examples showing several typical cases.

Up and Running

Finally, after a long delay in obtaining the Teletype (not purchased from MITS, but leased from RCA in New Jersey), the system was fully operational. We have been using it continuously ever since.

MITS is to be congratulated for their excellent software. Their version of BASIC is superior to most others that we have encountered, and it uses only 6 K of memory, allowing us to write programs of considerable length (about 100 lines).

The Altair is kept powered up continuously from Monday morning to Friday afternoon to save wear and tear on the paper tape with the BASIC software; also, it would be too inconvenient to key in the bootstrap program and wait the 12 or so minutes it takes to load BASIC every day.

The security of the system is important because the Altair and the Teletype are both kept out in the same classroom. Because the computer is not very large and thus easy to steal, special precautions had to be taken. The Altair is attached to the cabinet by three screws through the bottom of its case; it is positioned close to the rear of the cabinet so the top of its case cannot be removed with a regular screwdriver. Also, a Plexiglas shield was placed over the bottom

MITS is to be congratulated for their excellent software.

The first addition planned for this system is conversion to a magnetic tape interface. Further in the future, we see the memory expanding to 12 K bytes, purchasing the Altair Floppy Disk System, and trading up to MITS Extended BASIC.

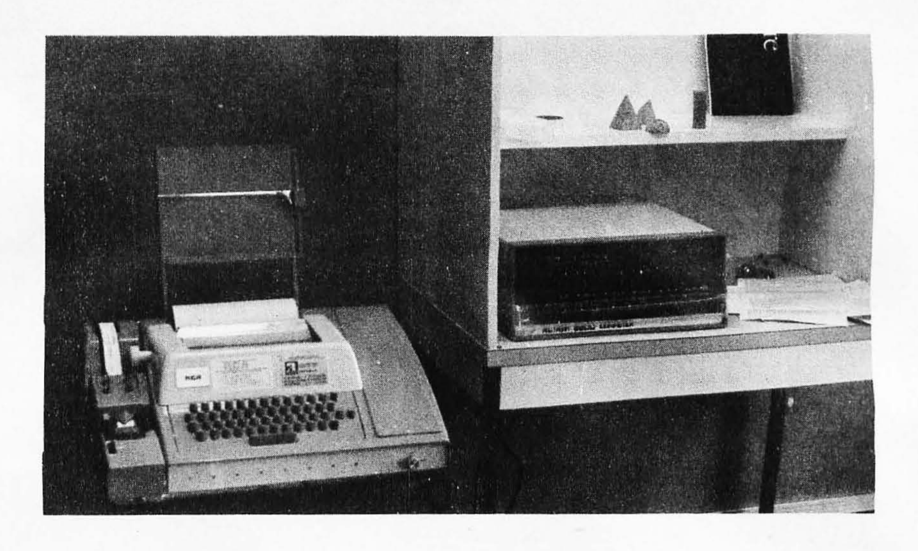


Photo 1: The John F Kennedy High School computing system. There's a Plexiglas shield over the bottom row of switches on the Altair. —**Photos by C T Nadovich**

of the Altair's front panel to keep anyone from inadvertently throwing the "OFF" or "RESET" switches and wiping out the BASIC.

A typical session at the terminal goes something like this: The student connects the computer and the Teletype by turning the selector switch to COM (communicate). There is, at present, no sign on or user password to control access to the computer. Making the student use the computer out in the open discourages those who are not authorized to use it from doing so. (Besides, 99% of the people who aren't supposed to use it wouldn't know what to do even if they got it powered up!) After making the connection, the student clears the memory to remove any data that may have been left by a previous user. He then either keys in his program or loads a program through the tape reader, and goes to work. He may also use a program from a library of special routines we have in several subject areas: mathematics, chemistry, and physics problem solving, lab simulations, text editing, and puzzles and games. When his session is finished, he can save the current program on paper tape, or simply clear the memory and turn off the Teletype.



Photo 2: Two students using the computer.

Future Plans

What does the future hold for this system? The first addition planned for it is the conversion to a magnetic tape interface. The MITS cost for this interface plus an additional expander board and a cooling fan comes to less than $170. The conversion will accomplish three important things: It will shorten the time needed to load BASIC from 12 to four minutes, provide for program storage in the more convenient form of tape cassettes, and it will allow us to trade down from the Model 33 ASR to the 33 KSR. The advantage of trading down is that the KSR leases for $15 per month less than the ASR since it does not have a paper tape punch or reader. The savings will pay the cost of the cassette interface in about a year.

Further in the future, we see the memory expanding to 12 K bytes, purchasing the Altair Floppy Disk System, and trading up to MITS Extended BASIC language, which has double precision arithmetic, controlled format output, and disk files. A TV typewriter or other similar video terminal is also envisioned.

Although this article has focused on use of a kit computer as an economical system for a small high school with a tight budget, the savings outlined are applicable for schools anywhere. In the face of rising commercial computing costs, a homebuilt minicomputer such as the Altair offers an economical yet efficient alternative to commercial computing systems for schools. ■

Can anyone beat the Altair System?

We doubt it.

When it comes to microcomputers, Altair from MITS is the leader in the field.

The Altair 8800 is now backed by a complete selection of plug-in compatible boards. Included are a variety of the most advanced memory and interface boards, PROM board, vector interrupt, real time clock, and prototype board.

Altair 8800 peripherals include a revolutionary, low-cost floppy disk system, Teletype,™ line printer, and soon-to-be-announced CRT terminal.

Software for the Altair 8800 includes an assembler, text editor, monitor, debug, BASIC, Extended BASIC, and a Disk Operating System. And this software is **not just icing on the cake**—it has received industry wide acclaim for its efficiency and revolutionary features.

But MITS hasn't stopped with the Altair 8800. There is also the Altair 680—complete with memory and selectable interface—built around the new 6800 microprocessor chip. And soon-to-be-announced are the Altair 8800a and the Altair 8800b.

MITS doesn't stop with just supplying hardware and software, either. Every Altair owner is automatically a member of the Altair Users Group through which he has access to the substantial Altair software library. Every Altair owner is informed of up-to-date developments via a free subscription to **Computer Notes.** Every Altair owner is assured that he is dealing with a company that stands firmly behind its products.

After all, we didn't become the leader by messing around. Shouldn't you send for more information or visit one of our Altair dealers?

Altair Coupon

Please send me the following information:

- ☐ Your latest catalog and price list
- ☐ Software information package
- ☐ Please include a list of your dealers

NAME________________________________

ADDRESS______________________________

CITY__________________ STATE & ZIP__________



2450 Alamo S.E. Albuquerque, N.M. 87106

A Systems Approach to a Personal Microprocessor

Dr Robert Suding
Research Director for Digital Group Inc
PO Box 6528
Denver CO 80206

Even a casual glance through the BYTE, *Radio Electronics*, *Popular Electronics*, etc, advertisements and articles reveals a growing proliferation of microprocessor integrated circuits and completed units. Which of these is right for you? Here are some ideas to bear in mind while making your choice.

Why do you want a processor at all? Reasons vary greatly. Many find themselves intrigued by the "computer environment" around us, and the microprocessor has become a low cost entry point into "computers."

Several amateur computer newsletters list reasons for individuals becoming interested in microprocessors. Hams see them as a working piece of equipment for their radio station. Hobbyists see them as process controllers; everything from lawn sprinkler controllers to robots. Mathematical types find them usable to run BASIC, FORTRAN, APL, etc, for problem solving.

What are your future plans with microprocessors? This may become a very open question. However, some reflection in this regard may prevent you from making an initial, very expensive, mistake. If you only have a casual curiosity, don't spend a fortune. A definite growth plan indicates a need for more careful analysis.

Investment

Microprocessor kits vary from $100 to several thousand dollars. The lowest cost units are excellent for satisfying curiosity about microprocessing in general, or will allow machine code manipulations. Several thousand dollar systems are often designed for and purchased by businessmen and professionals for applications such as payroll accounting, text editing or name file maintenance. The most frequent non business personal system investment is probably in the $500 to $1500 range.

Change

If there is one constant that is already evident in this field, it is constant change. You are about to invest (or already have invested) a significant amount of money in a microprocessor system. Unless your curiosity is easily satisfied, the chosen system should be able to easily adapt itself to

evolutionary changes being constantly invented or stressed. For instance, every six to nine months (Virginia Peschke calls it the gestation period) a major architecturally different central processor integrated circuit is announced. A system which allows upgrading without total obsolescence can be a real savings for the serious hobbyist. It can be very frustrating to be stuck with last year's wonder while everybody else has the latest microprocessor system. Several layers of change seem to be occurring. The fastest change seems to be the microprocessors themselves. The power supply and cabinet, if adequately large, can be a relatively stable portion of a hobbyist's system. The major expense in substantial processor systems is the memory components. A wise investment in memory will result in a system with a good life expectancy. The IO components are often a stable investment, sometimes an evolutionary element. A high resolution TV monitor, a mechanical hardcopy printer, or a good ASCII keyboard can outlive several generations of microprocessors. Expendable IO, such as cassette systems, analog to digital converters, and discrete IO circuits have shorter lives, but are lower cost. With proper design an evolutionary change can represent only one fourth or less of your total hardware investment instead of 75 percent.

Independency

An evolutionary system is best designed by making its various components independent of each other, and interfaced to commonly accepted levels and lines. Memory boards are relatively stable system elements in this kind of design: Speed and power consumption, besides price, are important considerations. Slower or surplus memory integrated circuits may be an expensive mistake if you want to run your latest model central processor which has become much faster. The slow memory may result in unnecessary central processor wait states. IO is generally processor independent, but IO interfaces can be susceptible to obsolescence when they depend on a specific central processor design. If you want to switch processors, they may require considerable redesign. A system which consists

of easily plugable boards can represent a major cost savings if they represent independency at the board level.

Quality

Of course everybody has it. Don't you read the advertisements? However, look beyond the surface for key items, or your long run investment will make you wish that you had. Here are some mechanical and electrical considerations of packaging:

- PC Boards – Double sided epoxy, plated, with plated through holes.
- Connectors – Gold plated fingers.
- ICs – Factory Prime, not temperature fallouts, etc.
- Conservative access speeds. Every IC socketed.
- Small Parts – Close tolerances where needed.
- Power Supplies – Conservatively rated, overcurrent, overtemperature, and overvoltage protected.

System Architectural Variations

There are a number of approaches to small system microprocessor design. Each is satisfactory for certain people, certain applications.

- *Toggle Switches and Bit Lamps:* The first hobbyist oriented microprocessor designs, and many present systems, are based on switches and lamps. If the system is limited to this, programs are small; or it takes long periods to enter longer programs, and are very susceptible to entry error. The user is forced to think at the micro level, bit by bit. If the intention of the user is to gain intimate logic knowledge of the microprocessor only, this system design is very cost effective.
- *Numeric Keyboard and 7 Segment Readout:* The ease of entry of this type of system allows a substantial gain in programming system complexity. However, the user is still at the logical data operation level. In addition, the programmer is restricted to viewing only a single byte at a time, making operator effort for analysis proportionally high.
- *Teletype or Similar Hardcopy Devices:* These systems represent the next level of improvement, offering some significant advantages. They usually have some form of monitor in a ROM which allows the operator to type in code and helps isolate him from errors. The total program may be listed or printed on hardcopy. In addition, paper tape is usually available to provide an economical media for program storage and exchange.

  There are some trade-offs, however. New hardcopy machines cost $1,000 up. Being mechanical devices, they require significant precision maintenance. The input/output speed is usually about ten characters per second; a dump of 1 K takes about two minutes, and creates a great deal of irritating noise. In addition, paper tape is a damage prone and bulky medium.

  Several integrated circuit manufacturers offer Teletype-oriented "evaluation boards." If only required for evaluation, ok; but they offer almost zero chance for either updating or extending. Both memory and IO are typically very CPU dependent, and if memory buffering is not used, supplemental memory and IO may be unusable.
- *Video and Cassette:* The latest stress has been the movement to using a TV set as an output display, a full alphanumeric keyboard for input, and an audio cassette for program storage and exchange. Video-based systems provide full user to system interaction at minimal cost. A *complete* video display and cassette based system will cost less than a hardcopy device alone. The speed of system response is practically instantaneous. Operations may be performed in almost complete silence (a major advantage to the hausfrau)! Reliability is enhanced as electromechanical mechanisms are limited to the keyboard and cassette recorder. Data media storage density is much higher; you can store the data from almost a mile of paper tape on a single C-90 audio cassette.

Conclusion

Serious hobbyists should carefully consider design alternatives and growth plans before ordering or building a microprocessor. Ease of operation, reasonable cost, and relative freedom from total obsolescence should be prime considerations.

In the following months, a detailed series of Digital Group hardware designs will be presented for your use. Next month will feature the low cost Digital Group cassette interface circuit which design provides data rates as high as 1100 baud, and may also be used as a ham RTTY terminal unit or as a telephone modem.■



Don Alexander of Columbus OH was named the grand prize winner in the MITS World Altair Computer Convention demonstration contest. Don is shown here with his computer controlled amateur radio Teletype station. The radio equipment is the stack at the left. At the right is his homebrew video display output on an OEM monitor, and in the background at the right is his Altair 8800 on top of which is perched the radioteletype converter unit. The station is completed by the keyboard unit in the front, and a Model 19 Teletype machine (only a corner is visible at right) which logs station contacts during contests. — **Photo by Andrea Lewis**

The Albuquerque Happenings



To our complete surprise, BYTE received an award at the MITS convention, prepared and delivered by a large SCCS contingent. This photo shows BYTE editor Carl Helmers (left) receiving the SCCS "special award" from Lou Fields of SCCS (right) while Dave Bunnell of MITS watches. — **Photo by Robert Prati**

At the recent World Altair Computer Convention, BYTE was represented by editor Carl Helmers, publishers Manfred and Virginia Peschke, circulation manager Deborah Luhrs, and Tully Peschke. The convention was capably organized and carried out under the direction of David Bunnell. Approximately 700 enthusiastic people from places as far away as Iran, Austria and West Germany came to Albuquerque NM for what was effectively a giant computer club meeting. Southern California Computer Society was present and accounted for with a chartered plane flight and many of the people who are responsible for that organization's existence. One of the major activities of the convention was a computer systems

demonstration contest. Judges were Les Solomon, technical editor of *Popular Electronics*, Larry Steckler, editor of *Radio Electronics*, Theodor Nelson, author of *Computer Lib/Dream Machines*, and BYTE editor Helmers.

Don Alexander of Columbus OH was named grand prize winner in the demonstration contest with his computer-controlled amateur radio Teletype station. The home built system consisted of an Altair 8800 with 8 K of memory, an ASCII keyboard, a video display, Baudot Teletype and standard transmitter and receiver.

In addition to building the hardware, Mr Alexander developed his own software and wrote the assembler and editor for the system. The program he demonstrated at the Convention was written for receiving and transmitting messages in a radio Teletype contest. The Altair 8800 kept track of most of the radio Teletype contest "housekeeping," such as: ASCII/Baudot translation, crosschecking calls for duplication, sending the time and message number of a transmission along with lines of text that are generated by command from the keyboard. After every exchange, a log entry was printed on the Teletype, keeping a hard copy record of all exchanges. A complete Altair floppy disk system was awarded to Mr Alexander for his winning entry.

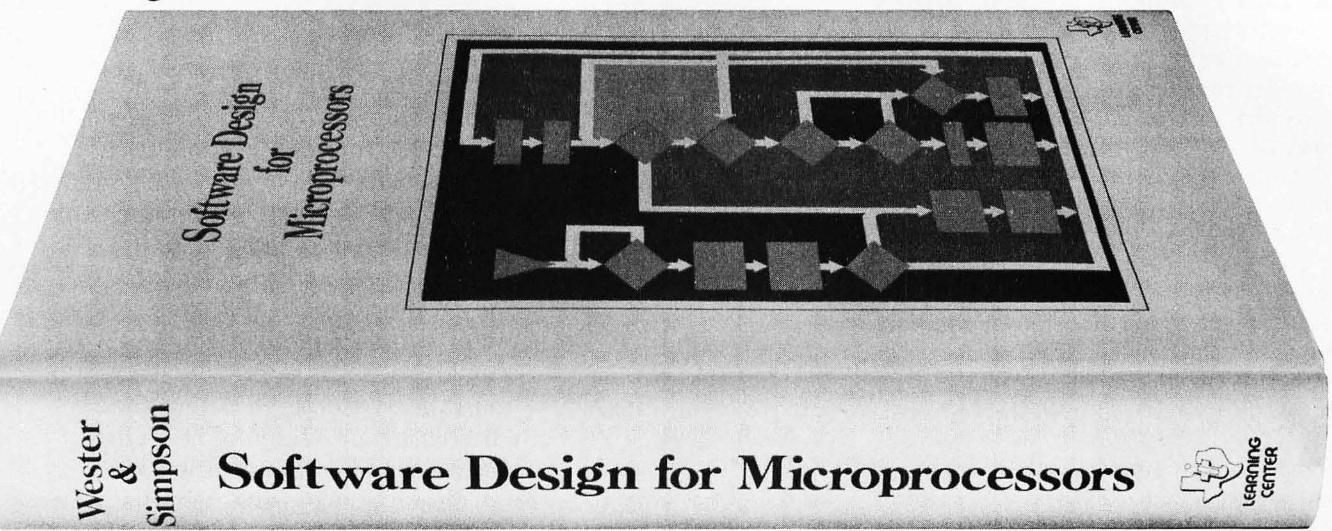
A tie for second place resulted in MITS awarding two Altair 8800Bs: one to Randy Miller of Tempe AZ for his computer chess demonstration; and one to Wirt and Valerie Atmar of Las Cruces NM for their speech synthesizer.

Third prize, an Altair 16 K static memory card, went to Danny Kleinman, Steve Grumette and Mike Gilbert of Los Angeles CA for their backgammon game, written in Altair BASIC and played on a Cromemco TV Dazzler.

The winners were announced March 28 1976 at the Altair awards banquet. Grand prize winners in MITS' yearly software contest were also named at the banquet; James Gerow of Houston TX won first place for the best program, and Lee M Eastburn of Langdon ND took the top prize in the subroutine category.

Throughout the weekend Altair users from 46 states and seven foreign countries took part in seminars presented by MITS engineers and software developers. A group of guest speakers, including Ted Nelson, author of *Computer Lib*, David Ahl of *Creative Computing*, and Carl Helmers of BYTE, led a Saturday night discussion of what the future holds for computers in general and for hobbyists in particular.■

New from Texas Instruments: An authoritative guide to understanding microprocessor software...from the beginning. Only $12.95.



A working knowledge of microprocessor software is essential to mastery of microprocessors. And acquiring such knowledge is now simplified with TI's new text, *Software Design for Microprocessors*.

This new book is designed to help you to fully understand the basics of microprocessor machine code and assembly language. It is equally suitable for the non-technical professional as for the technically trained.

Complete, stand-alone guide: Having more than 500 pages, the book contains the information you need to learn the language, special terms, and underlying concepts of microprocessors that lead to an understanding of the complex facets of software. Here are clear, readable discussions as well as references, graphs, tables, data sheets, examples and appendices. It's a convenient, single source to show you how to program microprocessors to do what you want.

Comprehensive text: In the first chapter, you learn basic terms, study basic machine architecture and look in detail at instructions as well as addressing. In succeeding chapters, you progress on to investigating the process of generating software...defining the support and documentation required...designing a simple machine to demonstrate how to program a problem. In conclusion, you gain experience by examining four sample problems.

This book is an ideal companion to TI's series of μP Learning Modules—learning systems to help you become familiar with fundamental hardware/software relationships. *Software Design for Microprocessors* is economically priced at only $12.95. Use the coupon to order your copy today.

LEARNING CENTER

TI Learning Center BY-4
P.O. Box 5012, M/S 54
Dallas, Texas 75222

Please send me______copies of (LCB 1891) **Software Design for Microprocessors** at $12.95* per copy.

I enclose ☐ check ☐ money order for $______.

☐ I would also like a brochure on TI's Microprocessor Learning Modules.

Mail checks and money orders to: P.O. Box 3640, M/S 84, Dallas, Texas 75285. Postage paid. Add state and local taxes where applicable.†

Mail company purchase orders to: P.O. Box 5012, M/S 84, Dallas, Texas 75222. Postage and taxes will be added to your invoice.

NAME

TITLE

FIRM

ADDRESS

CITY STATE ZIP

*Subject to change without notice

† AL, AZ, CA, CO, CT, FL, GA, IA, IL, IN, KY, MA, MD, MI, MN, MO, NC, NJ, NM, NY, OH, PA, TN, TX, UT, VA, WA, WI.

TEXAS INSTRUMENTS
INCORPORATED

61002

Building an M6800 Microcomputer

Bob Abbott
Route 4, Box 583
Evergreen CO 80439

Photos by Sheldon Luper

Early in the introduction of the Motorola 6800 system, an "Evaluation Kit" of seven family chips was made available. This kit contains the microprocessor, a read only memory preprogrammed with a system monitor called MIKBUG, two 128 by 8 bit random access memories, two peripheral interface adaptors and an asynchronous communications interface adaptor. These chips are all M6800 family members, intended to work together and demonstrate a typical minimum microcomputer system. At this writing the kit cost is about $150.

Because a number of these kits were and still are being purchased by hobbyists, I felt a few notes on my own experiences in getting one up and running might be helpful.

Construction

System layout and design was deliberately kept simple without compromising quality. Most components are on three Vector Inc 4066-5 wire wrap cards. These are extremely versatile plug-in boards having good power distribution and an excellent ground plane. The first card (see figure 1) contains the processor circuit, system clock module (Motorola MC6871A), MIKBUG ROM, and the system reset circuit. Also included was a socket for a 512 byte erasable programmable read only memory for my own firmware. I decided to include a total of six programmable random access memory integrated circuits to total 768 bytes, of which 74 bytes are dedicated to MIKBUG. These are mounted on their own card, connected as shown in figure 2. A third card (figure 3) is for input and output. It includes the two peripheral interface adaptors, the asynchronous communications interface adaptor, a bit rate generator, an MC14536 programmable timer used for MIKBUG and the RS-232 serial interface buffers.

All boards were wire wrapped using #30 gauge Kynar insulated wire and a modified wrap which includes one turn of insulation on each wrap for mechanical durability. Wire wrap was selected for this system in lieu of printed circuit boards because of the ease of changing connections. Total wrapping time for the three boards was about 10 hours.

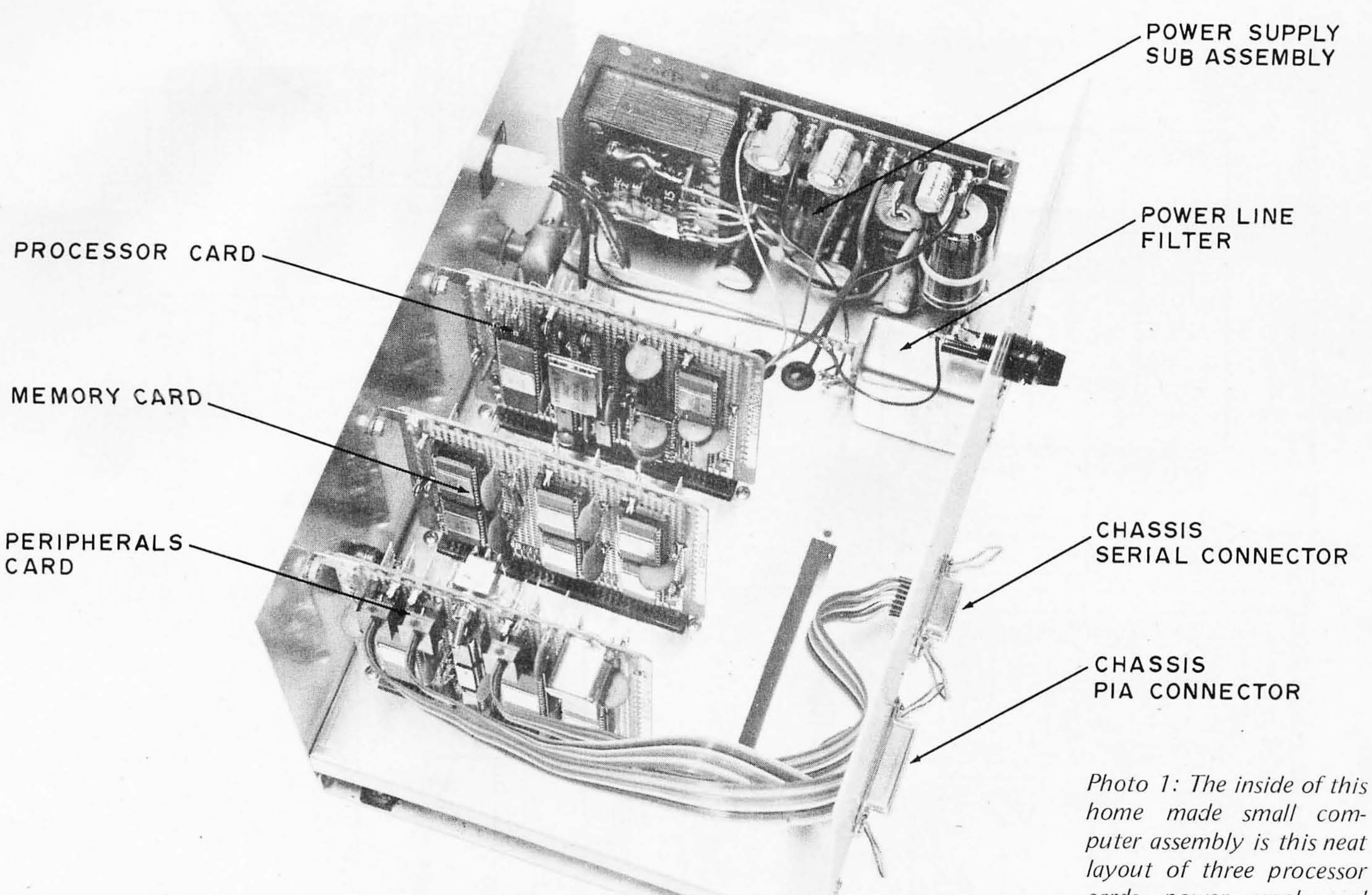


Photo 1: The inside of this home made small computer assembly is this neat layout of three processor cards, power supply and related components. The empty hole in the chassis base allows for one more card to be added at a future date. Wiring of the backplane is done on the underside of the card connectors. Note that only the MIKBUG ROM was plugged in when the photo was taken.

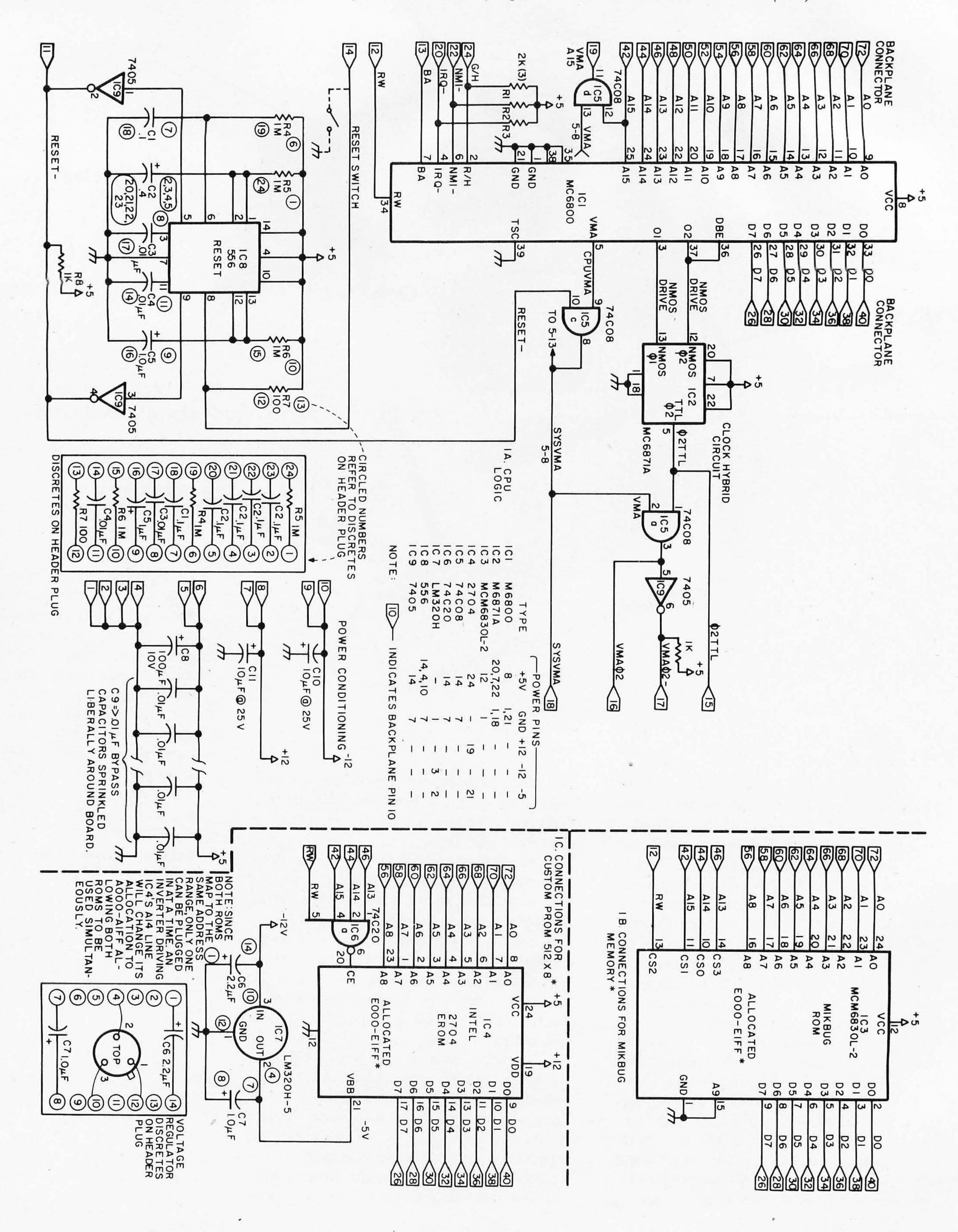
I decided to use a commercially available power supply, a Power One model HTAA-16W, because I did not feel I could conveniently duplicate the features of this unit for less than its $50 price tag. This supply provides 5 volts at 2 A foldback current limited and over voltage protected plus ± 12 volts at 200 mA foldback current limited. The HTAA-16W provides an adequate operating margin for the present system and remains quite cool even in a hot ambient environment. In fact the entire system dissipates little heat; the case does not even get warm to the touch. Photo 1 shows the physical arrangement of the boards and power supply for the system.

In keeping with the concept of simplicity there are only two controls on the front panel, a low priced keyswitch for power on-off and a master system reset push button (see photo 2). The keyswitch provides security from unauthorized tampering, the reset push button allows an exit from the target program back into MIKBUG. I considered adding a single step mode of operation but decided to delete that item due to having MIKBUG in the system. However, the additional circuitry is minimal and is detailed in the *M6800 Applications Manual* if you feel this option necessary.

I used a power line filter in this design primarily because the unit may be operated on lines shared with large motors. The power line filter is the large metal can attached to the rear of the enclosure near the fuse holder (see photo 1). I have enough problems with my software without elevators and such confusing things even more. All IO was brought from dual inline sockets via ribbon cable to the output connectors on the rear panel. One connector is the serial IO, the other is for the two parallel IO ports.

The main chassis carries the power supply and the three board connectors plus room

Figure 1: Processor Card. The major components are the processor, IC1, the clock generator, IC2, reset debouncing logic of IC8, and sockets for either the MIKBUG ROM (IC3) or an Intel 2704 programmed with an alternate monitor (IC4).



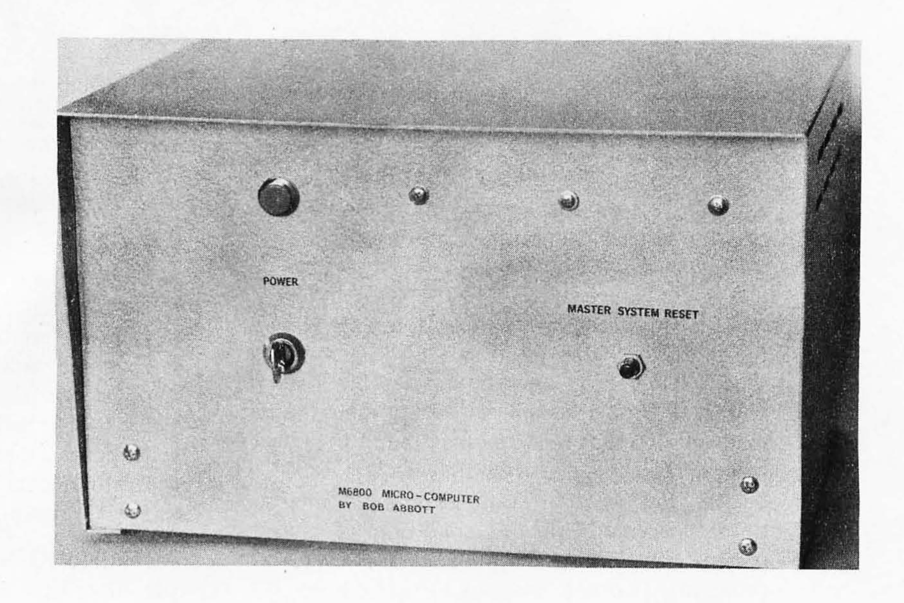


Photo 2: Corresponding to the simplicity of the interior layout, the front panel arrangement of this Motorola 6800 implementation has only two controls: A "Master System Reset" button for S1, and the power keyswitch, S2.

for one expansion board. All bus connections except power were wire wrapped; the +5 volt bus is 16 gauge stranded; the +12 volt and -12 volt buses are 24 gauge stranded; and ground is strapped to the chassis. Each board was buzzed out for wiring errors and each power pin on all sockets checked for correctness prior to insertion of the chips.

Some precautions were followed when handling the integrated circuits, which can be damaged by static electricity. They were kept in conductive foam until used; and when inserting them in their sockets, I used a grounded wrist strap plus a shorted edge connector on the boards. These integrated circuits are extremely difficult to plug in without bending leads, but with patience all were finally in place.

Bringing It Up

Hooking up my SWTPC CT-1024 terminal to the little computer and powering up was a bit disappointing, as unfortunately there was no response. Troubleshooting this little system was surprisingly straightforward and in the following paragraphs I will try to explain the problems I encountered bringing it up.

First analysis of the clock module outputs showed both processor clock phases were well within the specs. I had, however, overlooked the fact that the Motorola clock module system phase two output leads the processor phase two output by about 30 ns. This is a nice feature for systems involving significant propagation delays on this line. The reason is that all data transfers take place on the fall of the phase two clock pulse. Data from the 6800 processor is only guaranteed good for 10 ns after the fall of the phase two input which normally also drives the data bus enable (DBE) input. If system phase two goes through enough gates, the total cumulative propagation delay could well mean that this strobe shows up after data becomes invalid. This problem has been nicely provided for by the designers of the clock module in allowing this 30 ns grace period. I had originally tied DBE to system phase two to avoid loading the MPU clock line, thus effectively negating that advantage. The cure was to move the DBE back to the processor's clock line. It should be noted that in some systems a longer data hold time may be required even with the leading system phase two. An example of this would be a system driving slow memory through bus expanders. In this case DBE may be held high after the fall of MPU phase two, stretching the valid data time. Also in passing I would like to mention the importance of keeping the processor clock drive lines as short as possible. The 6800 processor is sensitive to clock pulse ringing; and due to the high capacitive load of its clock inputs, long inductive clock lines may add up to a real system problem. So these lines must be kept short, less than two inches if possible.

To continue with troubleshooting, check-

ing the output of the valid memory address (VMA) buffer showed considerable noise from the system phase two clock which was riding on the VMA signal. This clock line was being gated with buffered VMA in the same package, a 74H08, and there was apparently coupling to the other gates. The result was a fair amount of noise on these gate outputs and on the processor's address line A15.

This situation was possibly aggravated by the fact that the 6800 processor is only specified to drive one standard TTL load plus 130 pF of capacitive loading, and the load on A15 was near the maximum. In any case I replaced the 74H08 with a CMOS 74C08, and noted an immediate and dramatic improvement in the condition of the system. The noise disappeared and everything then worked without further trouble.

MIKBUG

The system monitor comes with an Engineering Note #100 which describes the operation of the program and includes a complete assembly listing. This program provides the following functions:

- Load memory from keyboard or tape.
- Examine and change memory.
- Load to tape.
- Print the contents of selected memory.
- Examine and change the processor's registers.
- Go to user program.
- Evaluate a maskable interrupt.
- Evaluate a non-maskable interrupt.
- Set a breakpoint in the *user program.*

The MIKBUG Note does not mention a couple of critical points. First, *in order to*

Figure 2: Memory Card. The connections for memory are illustrated in this diagram. Six MC6810L-1 memory chips are used to implement a total of 768 bytes of memory. The present design does not decode all address bits, so use of large amounts of memory expansion will require some additional decoding logic in the chip select lines.

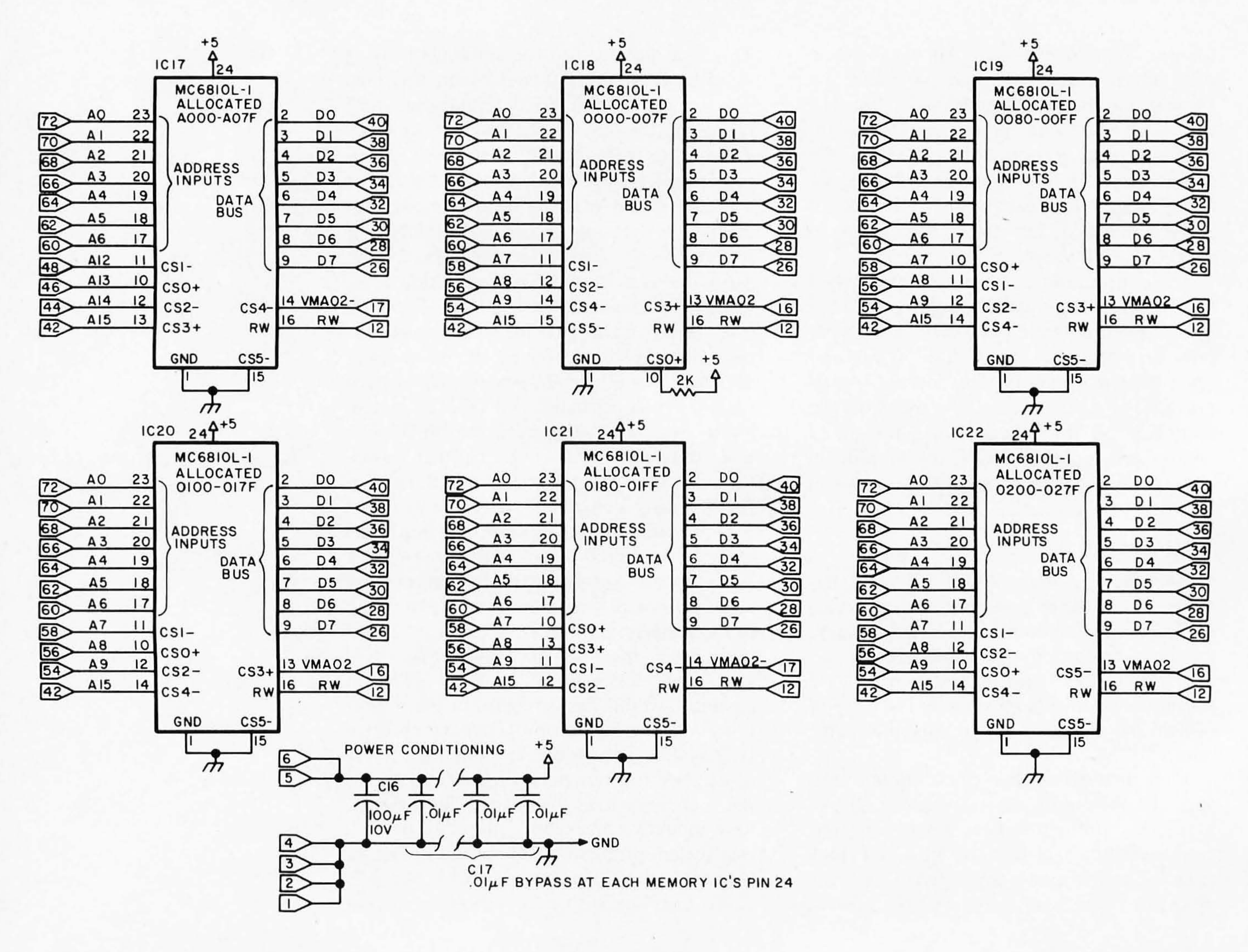
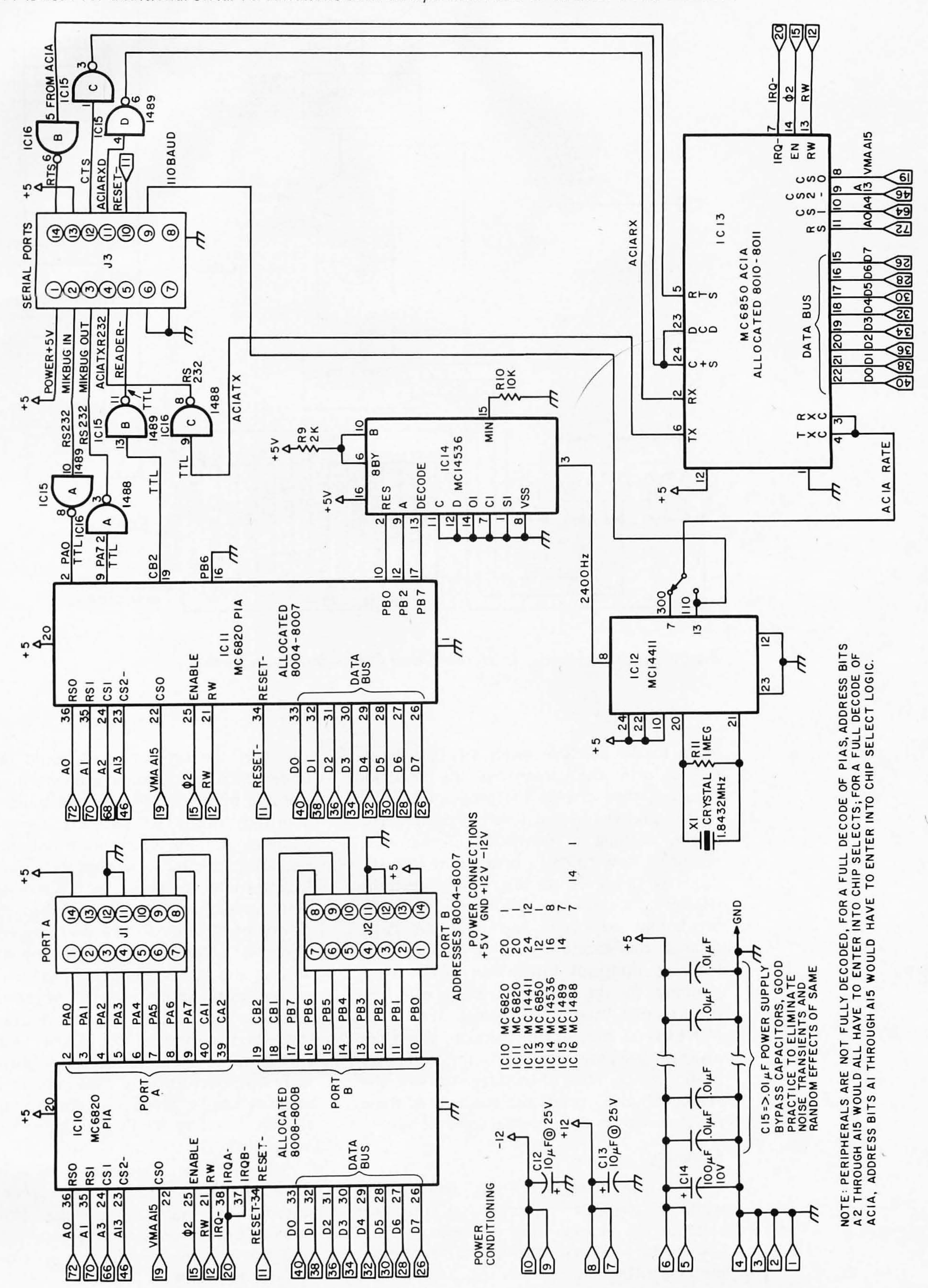


Figure 3: Peripherals Card. The system includes two PIAs and one ACIA mounted on the peripherals. The PIA at addresses 8008-800B is a general purpose IO port brought out to the PIA chassis connector via the two ribbon cables which plug into J1 and J2. The PIA at locations 8004-8007 is dedicated to the MIKBUG program's IO requirements. The ACIA at addresses 8010 and 8011 is used for additional serial IO functions such as tape interfaces or another serial terminal.



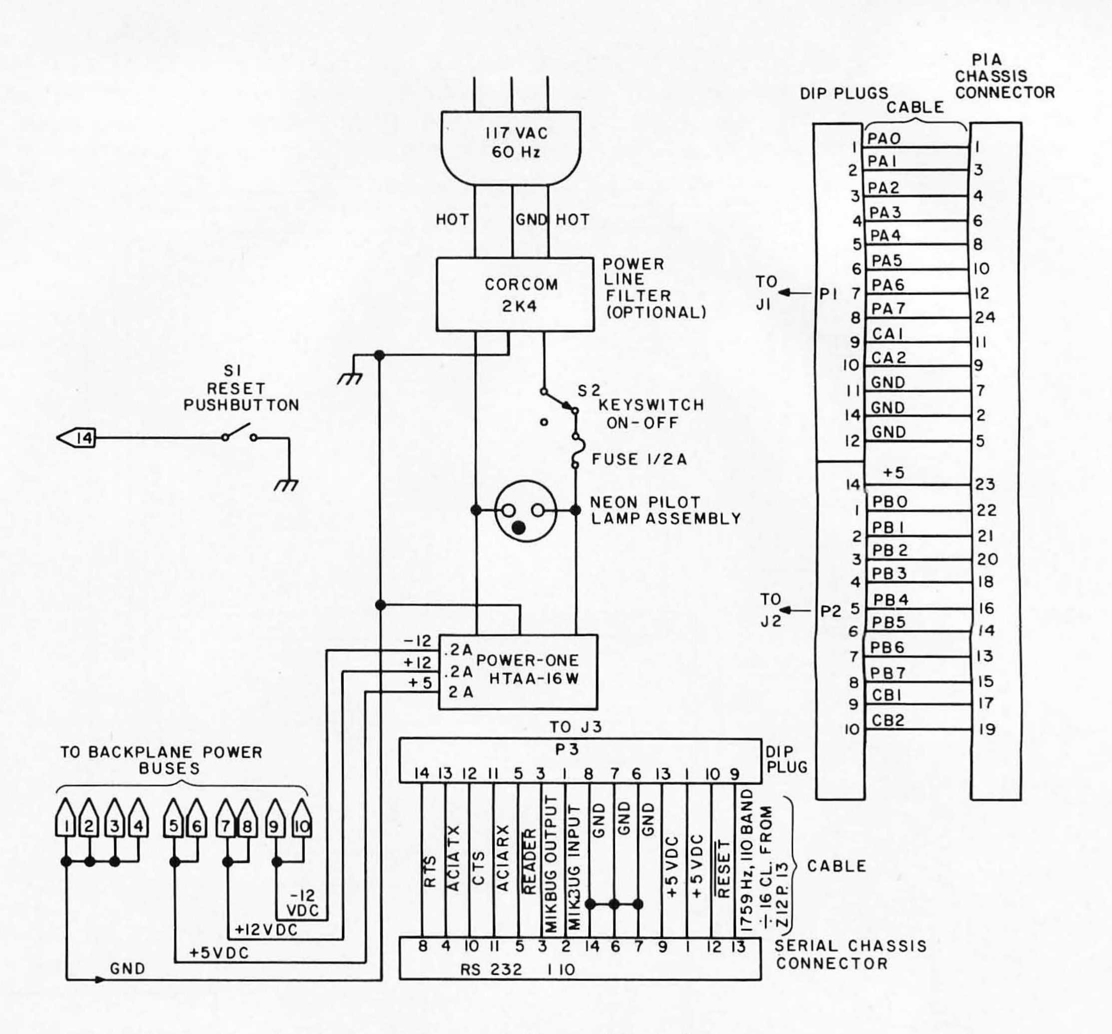


Figure 4: Power Supply, Interconnection Cables and Reset Switch.

jump to the user's program it is first necessary to print the contents of the processor registers, then change the program counter to the address of the target program. Secondly, nothing is mentioned in that note regarding how to set a breakpoint into the target program. To do this, simply open the memory location at which you wish the breakpoint to occur and note the data. Change this location to a 3F, code for a software interrupt instruction (SWI). Now, jumping to the target program with the breakpoint "trap" will cause return to MIKBUG at the SWI instruction. When the program executes the SWI, the computer pushes the contents of CPU registers into the stack; MIKBUG prints the contents of these registers and resumes normal control.

Summary

This microcomputer was constructed with an absolute minimum of time or expense. Troubleshooting and bringing the system up was straightforward, requiring minimal effort.

I feel the project's cost could be held under $250 with some prudent shopping. For the hobbyist on a limited budget, this approach might be the way to go. The addition of a TV typewriter produced a real operating computer system complete and lacking only somewhat in the area of random access memory space. Even so, there is plenty of memory for the average beginner. There is a considerable effort involved in writing and hand assembling programs long enough to fill all the available programmable memory. By the time the user reaches that stage of expertise he could start using his EROM to compensate for lack of programmable memory. In any case random access memory can be added with due attention to address decoding details as program requirements grow.

I hope these notes are of some aid to those hobbyists already owning or considering the purchase of the Motorola M6800 Evaluation Kit. I'm sure you will find this system as interesting to construct and use as I did.■

Why pay $3,000.00 for a source listing of a higher level microcomputer language?

SCELBAL is here. Now. Only $49.

Introducing the new microcomputer language that's simpler than machine language . . . **SCELBAL. SC**ientific **EL**ementary **BA**sic **L**anguage for 8008/8080 systems. This complete, illustrated program book . . . a gold mine for routines, techniques, source listings, flow charts and more . . . took several years to develop. Now it's yours for many years to come. The first time that intimate details of a higher level language program has been offered to the public at such a low, low price.

SCELBAL
A HIGHER LEVEL LANGUAGE FOR 8008/8080 SYSTEMS

Look at all you get! **5 Commands:** SCR, LIST, RUN, SAVE, LOAD. **14 Statements:** REM, LET, IF . . . THEN, GOTO, FOR with STEP, END, INPUT, PRINT, NEXT, GOSUB, RETURN and optional DIM. **7 Functions:** INT, SGN, ABS, SQR, RND, CHR, TAB. And, it runs in 8K and more.

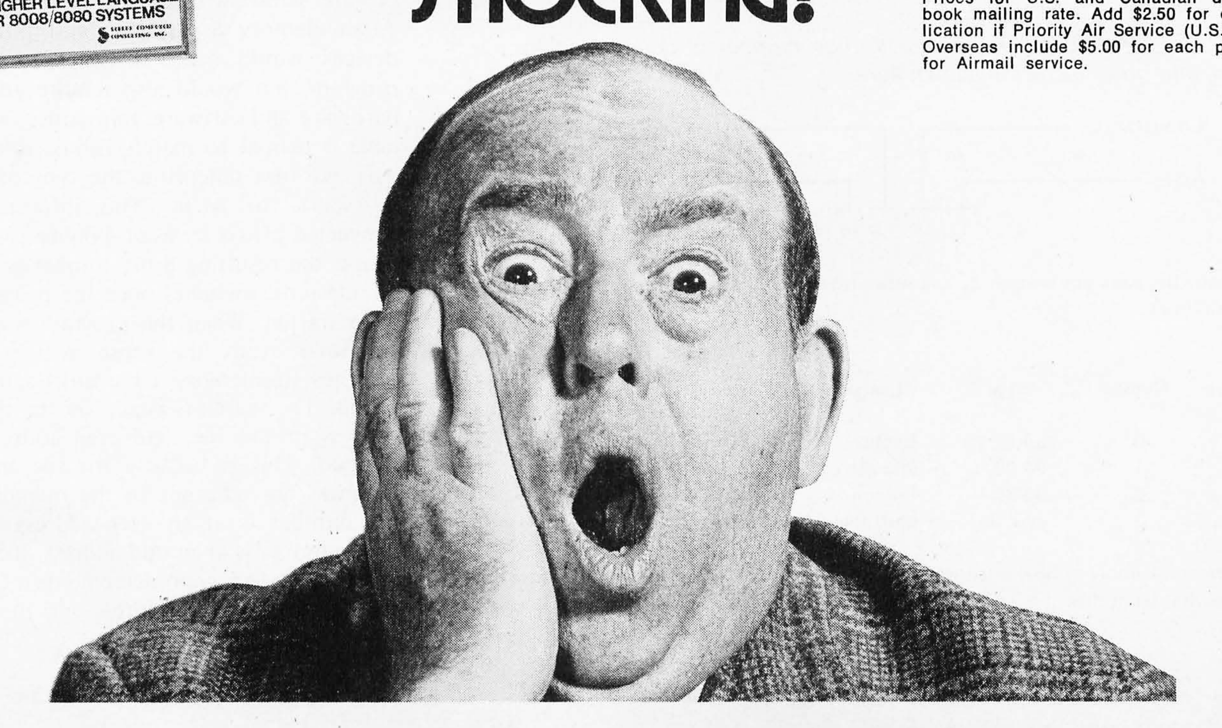
You read right! This book is yours for only $49 ppd. Now you'll have all the data you need to custom-tailor a high level language for your system . . . at a fraction of what it might cost until now. The program alone is worth more than its price. The revealed secrets of its operation are virtually priceless. Order today! "SCELBAL — A Higher Level Language for 8008/8080 Systems." Only $49 ppd. Delivery in June.



SCELBI COMPUTER CONSULTING INC. 1322 Boston Post Rd. Rear
Milford, CT 06460/(203) 874-1573

SHOCKING!

Master Charge, Postal and Bank Money Orders only accepted. Pricing, specifications, availability subject to change without notice. Prices for U.S. and Canadian delivery at book mailing rate. Add $2.50 for each publication if Priority Air Service (U.S.) desired. Overseas include $5.00 for each publication for Airmail service.

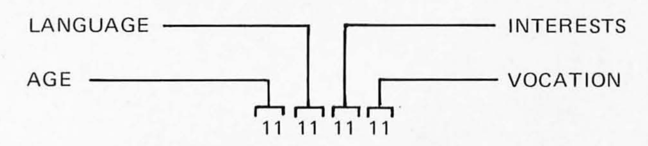


Strike a MATCH

Phillip L Hansford
6841 Haywood St
Tujunga CA 91042

Although this article is written from the standpoint of a MITS Altair computer, what is said here is also applicable to other systems. The original idea was simple: Using only a basic Altair (which started with 256 words of memory) and no peripherals, build a program which would match penpals according to age, vocation, and interests. It was originally experimental, written just to see if it could be done. But the program has proved so practical that we have been using it continually for our penpal club. The original program has been expanded to select compatible penpals from more than 200 choices. It has application wherever it is necessary to match data. In the version described here, a simple executive program can select the matching program as well as several other programs located elsewhere in memory. The executive reads the Altair sense switches for its inputs.

Table 1: For the penpal matching, a code number is determined by age, vocation, and other factors as shown here.



The four bit pairs are written as a number from 0 to 3. Each bit pair specifies one of four possibilities.

Bit Pattern	Number	Age	Language	Interests	Vocation
00	0	under 23	English	stamps	professional
01	1	24-29	Spanish	travel	worker
10	2	30-35	French	arts	student
11	3	over 35	Multilingual	other	other

This shows an example of how selection might be made. The actual categories in use now vary somewhat from this.

Penpals

Each penpal has an octal file number, and an octal code number which specifies several characteristics such as age, vocation and interests. Therefore, two words of memory are required for each person. The original version of the program gave us a capacity of 80 people; but when we added additional memory, the table area was expanded to hold 208 people. You could add even more memory to the program if desired, but in our use we did not need it; and the file numbers are 8 bit octal which limited us to 255 non zero names. (For uniqueness, file numbers should not be used more than once.) For this program, file numbers cannot be zero (which is interpreted as no answer); and interest code numbers can be neither zero (space), nor 377 (the stop byte).

Since there are presently no peripherals in the system, the information must be entered on the sense switches and read out in the Altair memory display (LEDs). Input output devices would make it easier to use the program, but would also require additional hardware and software. In practice, when we have a penpal to match against other penpals, we first determine the type of person he wants to write. This information is converted into a series of 4 digits using table 1 and the resulting 8 bit number is entered on the sense switches once the program has been started. When the program is run, the computer reads the sense switches, then searches its memory table until a match is found. The search is begun on the first odd address in the file, and even addresses are skipped. This is because the file and code numbers are adjacent in the memory. The file number is at an even address and the code number is at an odd address. If a match is found, the program decrements its address pointer to the even address and moves the

file number to the next available output location. It restores its pointer to the odd address and then increments the pointer twice to the next odd address and compares once again. If a match is not found, it increments the pointer twice to access the next record. The sequence continues until a 377 stop byte is found, which indicates that the end of the data table has been reached. At that point the program then jumps to a loop near the output addresses. The computer can then be stopped. The entire operation of the program takes the proverbial wink of an eye. The examine next switch is then used several times until the first output address is reached and the answer read on the panel LEDs. The next answer is available at the following address (depress examine next again) and so on. A zero in the output memory location means there are no more answers (or that there never were any).

The whole program is fairly simple, but we had to add a few touches of finesse before it would work properly. The output list of matches has ten locations allocated at the top of the 512 word region occupied by the program. This is adequate for our use; but if you anticipate more than ten answers at one time, you should modify the program to expand this space. The previous answers

Listing 1: The MATCH program specified in symbolic assembly language form, with absolute code for the program, starting at location 002/000 in memory address space. The notation A(X) is used to indicate the address of X. DS is used as a pseudo operation for reserving storage. DB is a pseudo operation used to indicate definition of constant bytes.

Intelese Octal Address	Octal Code	Label	Op	Operands	Commentary
002/000	041 364 003	MATCH	LXI	H,A (OUTPUT)	point to output area;
002/003	042 056 002		SHLD	POINTER	save pointer address;
002/006	257	CLRLOOP:	XRA	A	clear accumulator;
002/007	167		MOV	M,A	move zero to memory;
002/010	043		INX	H	increment memory pointer;
002/011	175		MOV	A,L	test low order of memory
002/012	376 377		CPI	377	against highest value;
002/014	302 006 002		JNZ	CLRLOOP	if not equal then continue;
002/017	066 307		MVI	M,307	place restart code in memory;
002/021	062 054 002		STA	CALLBUFF	set call buffer to 377;
002/024	000		NOP		left over NOP;
002/025	061 000 001		LXI	SP,A (STACK)	initialize stack pointer;
002/030	041 055 002		LXI	H,A (INBUF)	point to input buffer;
002/033	333 377		IN	SENSW	read sense switches;
002/035	167		MOV	M,A	save in input buffer;
002/036	021 115 002		LXI	D,A (TABLE +1)	point to first odd data entry;
002/041	257		XRA	A	clear accumulator;
002/042	276		CMP	M	is table code equal zero?
002/043	304 060 002		CNZ	MATCHER	if not then perform all tests;
002/046	303 360 003		JMP	DONELOOP	terminate the program;
002/051	000 000 000	DUMMY:	DB	0,0,0	unused space;
002/054	000	CALLBUFF:	DB	0	call buffer for executive;
002/055	000	INBUF:	DB	0	input buffer area;
002/056	000 000	POINTER:	DB	0,0	output pointer;
002/060	106	MATCHER:	MOV	B,M	B := INBUF;
002/061	032	REMATCH:	LDAX	D	A := TABLE [current odd byte];
002/062	376 377		CPI	377	is odd byte stop character?
002/064	310		RZ		if so then return to caller;
002/065	270		CMP	B	is odd byte equal match byte?
002/066	312 076 002		JZ	ITMATCHZ	if so then go process match;
002/071	023	NEXTREC:	INX	D	D := D + 1; [point to the
002/072	023		INX	D	D := D + 1; next table entry]
002/073	303 061 002		JMP	REMATCH	go retry with next entry;
002/076	033	ITMATCHZ:	DCX	D	point to name number of record;
002/077	052 056 002		LHLD	POINTER	point H,L to output;
002/102	032		LDAX	D	A := TABLE [current even byte];
002/103	167		MOV	M,A	M(POINTER) := A;
002/104	043		INX	H	POINTER := POINTER +1;
002/105	042 056 002		SHLD	POINTER	save POINTER in memory;
002/110	023		INX	D	point to odd byte again;
002/111	303 071 002		JMP	NEXTREC	go handle next record;
002/114		TABLE:	DS	416D	reserve 416 bytes storage;
003/356	000 377		DB	0,377	end of data area is set;
003/360	373	DONELOOP	EI		enable interrupts;
003/361	303 360 003		JMP	DONELOOP	and commence endless loop;
003/364		OUTPUT	DS	12	reserve 12 bytes storage
003/376	000	DONERST	NOP		NOP shows no answer;
003/377	307		RST	0	program restarts when incremented by hand;

also must be cleared each time the program is run. The first 24 bytes of the program were added to perform this initialization and some other housekeeping operations. A zero on the sense switch inputs is supposed to give no output from the program. We accomplish the test for this condition by placing the search and match in a subroutine and calling it if, and only if, the sense switches are not zero.

Expanding the Altair

The program shown in listing 1 is a final version of this MATCH program application which we concocted after a total of 1024 programmable RAM words was implemented on our Altair. This allowed us to make the data table accomodate 208 names with the program and data located at addresses 002/000 to 003/377. The lower 512 bytes of the memory were devoted to a simple executive program used to select which program to run, and space for other programs. The original matching program had an occasional bug: If you forgot to reset it before running, it might not jump to the output loop (the stack would overflow). Also, as mentioned earlier, more than ten matches was not acceptable and could conceivably cause additional bugs.

To overcome all this, the initial stack address was moved to location 000/377. Remember that with the additional memory, the match program was now moved to 002/000 through 003/377. Although the stack is allocated to addresses lower than 000/377, the stack pointer is initialized at 001/000, since the first location is ignored by stack operations. For ordinary uses, I allow at least 20 words for the stack. An enable interrupt was added near the end of the program (003/360) to coordinate system operation. An original halt at the end of

Listing 2: An EXECUTIV program for a minimal system. This shows a simple little program which is accessed via the Aux 2 switch interrupt and is used to read the sense switches and test for a particular program identification code. A flag called CALLBUFF (located in MATCH) is referenced to determine whether data input or a program selection is desired.

Intelese Octal Address	Octal Code	Label	Op	Operands	Commentary
000/000	000	MSTART:	NOP		this blanks display when stopped;
000/001	303 000 002		JMP	MATCH	RST 0 gets execution to MATCH;
000/004 to 000/067	} this area is open for arbitrary programming use;				
000/070	333 377	EXECUTIV:	IN	SENSW	A := sense switches;
000/072	006 377		MVI	B,377	B := 377 [timing loop constant];
000/074	016 377	OTLOOP:	MVI	C,377	C := 377 [timing loop constant];
000/076	015	ITLOOP:	DCR	C	C := C − 1;
000/077	302 076 000		JNZ	ITLOOP	if C NE 0 then repeat inner loop;
000/102	005		DCR	B	B := B − 1;
000/103	302 074 000		JNZ	OTLOOP	if B NE 0 then repeat outer loop;
000/106	107		MOV	B,A	B := A [save sense switch input];
000/107	072 054 002		LDA	CALLBUFF	A := CALLBUFF [continue if 0 else restart];
000/112	267		ORA	A	test A and set flags;
000/113	170		MOV	A,B	A := B [flags are unaffected];
000/114	310		RZ		if zero then return to program;
*000/115	376 xxx		CPI	P1	is program 1 symbol present?
**000/117	312 yyy yyy		JZ	PROG1	if so then go to that program;
*000/122	376 xxx		CPI	P2	is program 2 symbol present?
**000/124	312 yyy yyy		JZ	PROG2	if so then go to that program;
*000/127	376 xxx		CPI	P3	is program 3 symbol present?
**000/131	312 yyy yyy		JZ	PROG3	if so then go to that program;
000/134	267		ORA	A	is MATCH program symbol present (0)?
**000/135	302 yyy yyy		JNZ	PROG4	if not then go to program 4;
000/140	307		RST	0	otherwise call MATCH with restart;

Notes:

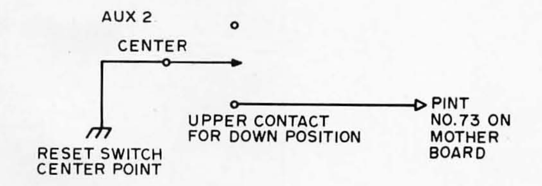
* "xxx" should be replaced with an arbitrary bit pattern. This becomes the symbol which identifies the given program, which will be input from the sense switches and matched in this little executive program.

** "yyy yyy" should be replaced by the address (low order first) of the program being accessed.

Any program which is to have provision for a restart should clear CALLBUFF and then issue an RST 7 to enter the executive. The executive will then read the sense switches as an input to the program and return with the input in the accumulator, instead of choosing another program.

The timing loop used to delay approximately 0.75 seconds is programmed according to the technique described by James Hogenson in "Can Your Computer Tell Time?", page 82, BYTE December 1975.

Figure 1: A hardware modification to the Altair 8800 which allows the Aux 2 switch to generate an interrupt. This modification is used to coordinate operation of the simple executive and inputs to the MATCH program described in this article.

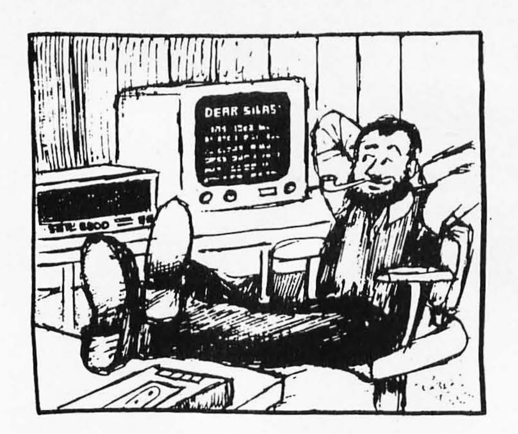


processing was changed to an RST 0, so you can run the program from an output location where threre is no answer if you like, rather than resetting the program. For the configuration shown in listing 1, the high address of the output list is the same throughout, so some provisions to change this at 002/017 through 002/024 were made into no-ops without affecting performance. It was then possible to insert instructions at 002/017 and 002/020 to load a 307 (RST 0) at the logical end of program execution. Since the no-op before it (at 003/376) was automatically cleared by the initialization part of the program, this meant that up to 12 answers could be accommodated and that if the program was reset, it would properly operate the next time. Note that if more than 10 answers occur, it is necessary to keep careful track of them so as to not mistake the restart as an answer. More than 12 answers will overflow the memory and be ignored in this configuration. If you have memory beyond 003/377, then the answers will keep on going unless you add a provision to limit their number. This is easily accomplished in about four bytes, but much of the program (and jump addresses) must be relocated to do this. A flag called CALLBUFF (002/054) was initialized to all ones using bytes 002/021 – 002/023, permitting system versatility, since the executive tests for it before selecting a new function. Space is available at 002/024 and 002/051 – 002/053 to accommodate program changes, or two additional data pairs may be added to the data table. (We didn't need to do either of these, so the available space is left for future changes.) The final form of the program is shown in listing 1.

The executive program shown in listing 2 drives our system. It makes use of an interrupt switch which was added to the Altair on the second auxiliary switch (see figure 1). This interrupt function is easily accomplished by running a jumper from the center terminal on the Aux 2 switch to ground (a convenient point is the center terminal on the reset switch) and another wire from the upper contact on the Aux 2 switch (for down position) down flat against the board and across to backplane pin #73 (PINT) on the mother board. Use the unused hole at the end of the mother board opposite the other wire connections. Be sure you locate the correct hole! This change will cause a program which has enabled interrupts to restart at 000/070 when the Aux 2 switch is depressed.

Although the executive program is very small and straightforward, it is effective and can direct the computer to any of five distinct locations (including the matching program). To operate in the executive configuration, run from address 000/000. The executive will automatically jump to the matching program and remain in the output loop until stopped or interrupted. To run a different program in the system, the appropriate code is now placed on the sense switches and the interrupt (Aux 2) depressed. There will be a delay of about 0.75 second, to give you time to release the switch, followed by the selected program. When the system is first run through the matching program, it initializes CALLBUFF to all ones. The executive tests this location to ensure this condition before it selects a new program. If all zeros are present at the call buffer, the executive will instead return to the previous (interrupted) program. Thus data can be input into a program (it will appear in the A register) with the sense switches and interrupt; just clear CALLBUFF (002/054) in your program. Also allow for the use of A,B, and C registers by the executive. Be sure that each program either resets the stack pointer or returns to the matching program at its completion. Otherwise the stack might get too large and overflow. Actually, this is all a lot easier than it might sound. A plain Altair computer is an exciting device and machine language can be easy to use. A small executive organizes it all together. What we have here is the beginning of a true system of hardware and software.■

Letters



HIGH LEVEL LANGUAGES

First let me congratulate you on such a fine magazine. I receive many different magazines and I like yours the most.

I would also like to congratulate you on trying to produce some standards in the hobbyist field.

I have noticed your disappointment in any high level languages offered for computer hobbyists and this brings me to write to you. I would like to get hobbyists' ideas of what would be a good high level language. I'm not guaranteeing anything, but if I can compile a list of what other hobbyists are looking for in a high level language, we (computer hobbyists) might be able to write our own standard language.

Of course, I have my own ideas and with all the different machines available, I think the only way a person could do it is to draw an *extensive* flow chart and let each person implement it on his machine.

The flow chart could cover a wide variety of things and if your system doesn't have some of the hardware necessary, then you would have to implement that portion.

Anyway, it's an idea and I would like to hear from hobbyists on what they think the high level language should be able to do.

Robert Sikes
Rt 5 Box 174
Biloxi MS 39532

IRAN INTO YOUR MAGAZINE THE OTHER DAY

As APO 09205 is in the mountains of Iran, I was not able to visit my local book store and purchase the previous issues of BYTE. If at all possible I would like to purchase issues #1 to #4, one copy of each.

Donald N Wagman
USAFE DET-5 ETU-3
Box 1600
APO New York 09205

WOULD-BE AUTHORS, TAKE NOTE

You may have many readers, who, like myself, have a good background in RF communications, digital logic, industrial electronics and test equipment. These are mostly of a hardware nature and when I look through BYTE some of the programing articles hardly resemble English. That's fine for those "in the know," but you need a series to take the hardware fans by the hand and lead them to software.

The equipment advertised in BYTE has a strong appeal to a hardware man who is not scared of chips, boards, interface gadgets and power supplies. The question is: After it's built, what then? I can't talk to it!

I think your readers and particularly your advertising customers would benefit from mass conversion of hardware people to the point that they *would buy an* Altair or SWTPC job with no fear that the fun would end with the last solder connection.

Think it over; the vast majority of electronics people are "hardware" at present; there is a fortune to be made!

Gordon D Stewart
Thompson, Manitoba
Canada

BYTE article SNOBOLs

I enjoy your magazine a lot, and especially liked the February issue article on processing algebraic expressions. Maurer's article was so interesting that I tried it out in a high level language (SNOBOL) and it worked great! My next step is an assembly language implementation for my M6800 and MCS6502 processors. I would really appreciate a similar approach on "hashing" or symbol table lookup.

Don Peters
Nashua NH

ANY AUDIO WILL WORK

As a present (or past) subscriber to several well known electronics magazines, and the receiver of five free-bee trade magazines (electronics) and three free-bee computer magazines, I was overjoyed to see the birth of BYTE. It has definitely filled a void!

My reason for writing this letter (besides patting you and your associates on the back) is to find out additional info on the BYTE standard for magnetic recording as it applies to reel-to-reel recording. I presently have a Teac tape deck which is capable of recording at 3-3/4 ips and 7-1/2 ips only. Needless to say, all the talk in the world about 1-7/8 ips

Establishing BYTE Committees of Correspondence

To encourage correspondence among readers, beginning with letters received after May 1 1976, BYTE will print the name and full address of each published letter's author. If you do not wish your address to be printed, mark it "do not print my full address" or the logical equivalent.

doesn't do me any good. I know that with higher speed I can get better throughput and better frequency response, but the exact way to adapt articles like Don Lancaster's "BIT BOFFER" and Harold Mauch's "Digital Data on Cassette Recorders" eludes me. I presently am trying to get a used TTY working and when I finish that, I intend to implement an MOS Technology 6502 microprocessor system. At this point I will want to implement some means of mass storage; and considering that the only available device I have is the Teac tape deck . . . Well, you get the idea.

James T Lareau
Parsippany NJ

The standard defined in BYTE's February and March issues as a result of the meeting we sponsored last fall might also have been named the "audio information exchange" standard. The two articles describe a way to generate and interpret audio wave forms. Any recording medium — such as your TEAC tape deck — which can faithfully reproduce the audio frequencies in the bandwidth of an inexpensive cassette deck could also be used with equipment built to the standard. Besides reel to reel tape, other media potentially usable include telephone channels, radio transmission of voice grade information, and phonograph records.

PORNOGRAPH RECORDS

When first your magazine I did see
I had no idea of what the contents could be.
A cover of blue titled BYTE
to me implied articles perverted authors would write.
Amidst other periodicals like Kill, Stab, and Maim
BYTE is a much needed change.
To cut this short and save some time
I would like to subscribe for three years time.
And please, kind sirs, charge this to BankAmericard No. XXXX XXX XXX XXX
because I haven't had money since I can remember.
In closing, one last request
please send all back issues you possess!

Steve R Burns
Ypsilanti MI

SAGE ADVICE

I would like to let anyone who is experimenting with building his own computer system or microprocessor know that I would like to communicate with him, swap ideas, software and hardware.

I am currently building up two micropro-

cessors using an Intel 8080 and a Motorola 6800. I would also like to swap or sell (for storage and shipping costs) an IBM line printer, card reader and tape drive that came off the famous SAGE Air Defense computers the US Air Force operates. I bought the machines when the Air Force closed its SAGE Air Base in New York. I also have some large scale computing facilities available which may be of interest to experimenters.

Milton Goodman
Techno cadimum Data Corp
101 Park Av, Suite 707
New York NY 10017

I THINK YOU BLEW YOUR COVER!

The cover of your December issue was a giant step in the wrong direction in terms of getting home computers accepted by the non-participating percent of the household. The gift recipient pictured is keeping the rest of the family away, the little girl behind him looks upset or worried (note the eyebrows) and the ever so attractively depicted wife could kill Santa with that look. Santa doesn't seem very confident. . . .

You should have considered the two-edged aspect of humor before giving a computer-wary family a blueprint in negative reactions this Christmas. Once the first good natured reaction to the cartoon drifts away, the shopper's buying power may not be used to purchase that ultimate toy.

I might note here that I think the CONTENTS of your magazine are first rate. The articles are interesting, well chosen, and well written. Topics are relevant to what's going on. On the whole, your magazine stands out, mercifully, in the oceans of newsletter material available, as a focal point of small systems information.

B L Donelan
San Diego CA

So far, yours is the only comment, one way or the other, on the subject of cover materials. Any others?

DISDAIN

What can I say to my wife after I tell her I want to "invest" our savings into a minicomputer and she says "What do you want with a computer?" with complete disdain. I need a short concise reply that justifies the expenditure as well as forever silences her on the matter. Hurry!

Charles Hurlocker
Seattle WA 98105

It's an "automated busywork eliminator."

UTILIZING SPECIAL CASES

Looking at the code for the 6800 caterpillar program (BYTE, March 1976, page 90) reminded me of some rules of thumb for the Motorola device:

1. The X and S registers (if not in use at the moment) are very handy for moving 16 bit fields around.
2. When using instructions which do not have direct addressing, space can often be saved by clearing X and using indexed addressing.
3. The CLR instruction often wastes space. Clearing A, B or X and then storing is usually better.

Applying these rules and one dirty trick gives the following version of the program:

```
1000 CE  3F  00  CATERPLR LDX  #$3F00
1003 DF  12               STX  R2
1005 DF  11               STX  R1
1007 DF  10               STX  R0
1009 0D                   SEC
100A CE  03  FF  NEWMOVE  LDX  #$03FF
100D 09          WAITLOOP DEX
100E 26  FD               BNE  WAITLOOP
1010 66  13               ROR  R3,X
1012 66  12               ROR  R2,X
1014 66  11               ROR  R1,X
1016 66  10               ROR  R0,X
1018 DE  10               LDX  R0
101A FF  01  4C           STX  LAMP0
101D DE  12               LDX  R2
101F FF  01  4E           STX  LAMP2
1022 20  E6               BRA  NEWMOVE
```

This version requires only 36 bytes versus the original 54.

One other very minor point. Since we are shifting a 7 bit caterpillar along the lights, the original program could be modified to eliminate the SEC instruction. Simply make the first instruction LDAA #$7F, putting all 7 bits in R0. It would then not be necessary to clear the carry bit because of the CLR instructions (which always clear C).

No doubt shorter versions of this program will be devised. My purpose in writing was to give a little boost to the X and S registers. Of course, a good discussion of the S register qua stack pointer could fill a small book.

George E Beine
President, Gnomon, Inc
Cincinnati OH

PS When is Motorola going to take a firm stand on the NOP? As one example of their vacillation, compare pages 3-2, 3-3 and 3-4 with page A-44 in their *M6800 Microprocessor Programming Manual* (second edition).

Excellent! Optimization is the art of reducing the memory required by a program, or the execution time required, or both. Optimization can be one of the most interesting challenges, especially when the task

becomes reducing code for an already tight routine such as the one printed here.

BYTE's policy with respect to programs submitted by authors (including those of a certain editor) is to inspect the accuracy of the code submitted with respect to accomplishing the tasks described by the article, but to avoid changing code unless a technical error is found. Thus in many of the examples of code printed in the magazine, there will no doubt be places where optimization could reduce memory requirements or speed up execution, as in this fine example provided by George Beine.

THE COMPUTER ARCHITECT'S SEARCH FOR UTOPIA

In a letter in the March BYTE, Michael A Sicilian writes, "One way around the problem of 8080 programs not working on a 6800 and vice versa might be to identify the 256 most valuable and powerful instructions we might desire in an 'ideal' 8 bit processor." Mr Sicilian has a truly wonderful and marvelous idea.

It is such a good idea that Intel has already done it. They call the result the 8080. Motorola has already done it and they call it the M6800. Advanced Micro Devices looked at the 8080 and decided it is almost right; they call their improved 8080 the 9080. When MOS Technology did it they called the result the 6501 (and the 6502 and the 6503). And Fairchild thinks the F8 has the most valuable and powerful instruction set possible. RCA says, "No, it is the CDP1801, COSMAC." There are more. Mr Sicilian would join this illustrious company. More power to him.

Unfortunately, when he has completed this tremendous labor of love, he will find that he has not replaced Babel with a world that speaks only one language. Alas, he will have added one more language to Babel. But pursue the effort. Once BYTE carries a description of this new best of all possible instruction sets, some manufacturer may well decide to immortalize it in silicon.

Daniel Lance Herrick
Owosso MI

HELP!

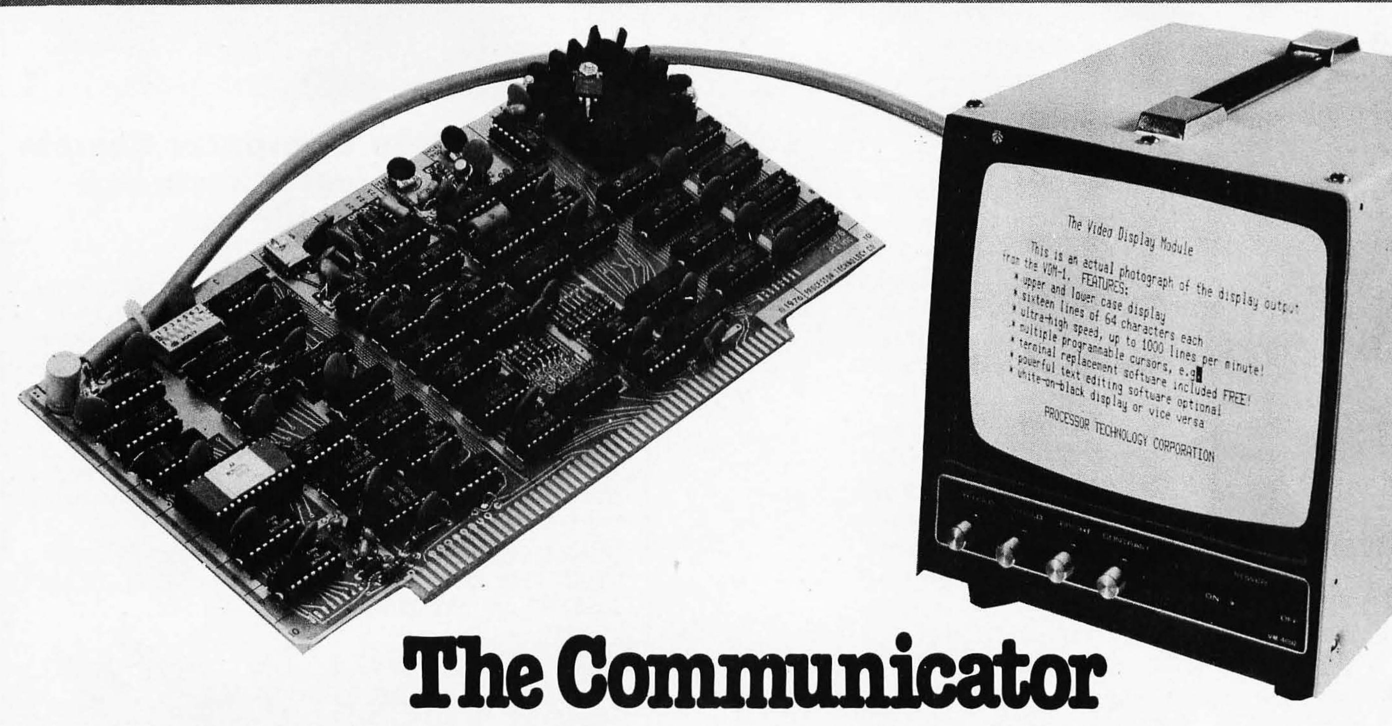
Who are suppliers of computers and equipment in Canada?

T M Atwood
1319 Stafford Dr N
Lethbridge, Alberta
Canada TIH2C3

We'd let you know if we knew. Can any readers help Mr Atwood?

The 8080 Expa

Let Processor Technology expand your Altair o and reliable plug-in modules



The Communicator

Your Altair already has the intelligence, so let our **VDM-1 Display Module** make the best of its capacity to communicate. This is **not** a limited "TV Typewriter." The VDM-1 is an **ultra-high speed output device, built right into your 8800 system.**

The VDM-1 generates **sixteen 64-character** lines in a large easy-to-read font with both **upper and lower case letters.** It contains 1K (1024) bytes of random access memory, to which the processor can read or write, just as though the memory were an integral part of the system. As the information is written in, contents of the on-card memory are displayed instantly without interrupting the operation of the processor.

Once the processor provides the display status parameters, the VDM-1 can be made to **"scroll" its display upwards or downwards.** A built-in timer allows scrolling at about 4 lines per second, eliminating complicated timing program routines. At top speed, the display scrolls through a dump of 65K of memory in two minutes; that's about 1000 lines per minute!

Multiple programmable cursor circuitry is built in. All 1024 cursors can be displayed at one time or begin anywhere in the display. Thus, the VDM-1 can display white-on-black or black-on-white—perfect for many video games! The VDM-1 also features EIA Video output for any standard video monitor, or a television repair shop can easily modify your own TV set.

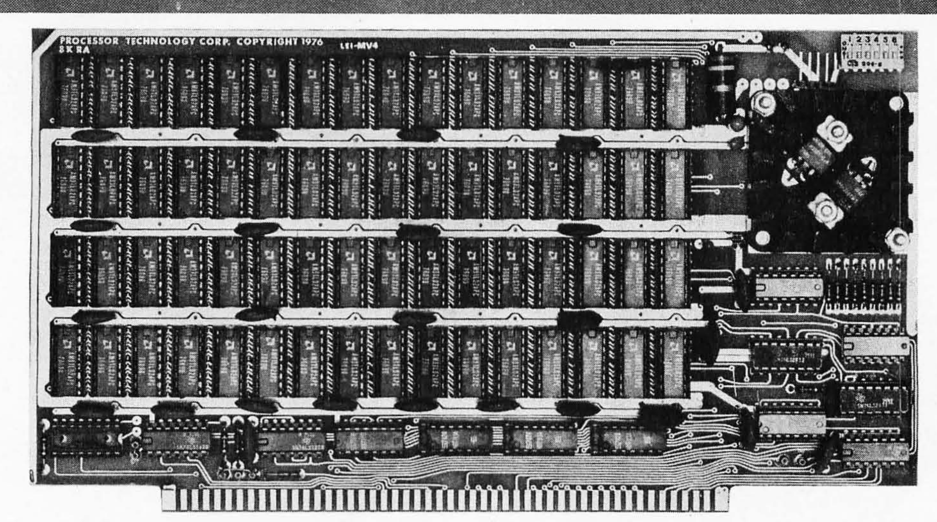
The VDM-1 comes with **free terminal mode software,** designed for teletype replacement when used with BASIC or our own Resident Assembly system. (Powerful text editing software and various game packages are also available from Processor Technology Corp.)

Our detailed VDM-1 Owner's Manual is available for $4.00, refundable with purchase of the VDM-1.

Kit Prices, $179 (premium grade, low profile IC sockets included).

nsion Principle.

IMSAI system—with innovative, excellent quality [At most agreeable prices, too!]



Sharpen your Memory!

Processor Technology's new **8KRA Static Memory Module** will add double-capacity, lower power, and greater efficiency to your Altair or IMSAI system. We've actually improved on our already popular 4KRA board, by doubling the capacity **(8192 eight-bit bytes!)** and lowering the power-per-bit.

The 8KRA uses full speed, low power static RAM's—our prime supplier is AMD (91L02A's). These RAM's are manufactured to stringent military specification, MIL STD-883A, to insure reliability. **The 8KRA uses less power-per-bit than any other true static memory module**—requirements so low, that two "D" flashlight cells will retain data for 4-5 hours. (Recharging circuitry for Ni-Cad cells and battery connectors are on the card.) Also, unlike dynamic memories, static memories do not require periodic refreshing, allowing them to run at the processor's **maximum speed.**

The 8KRA occupies any 8K segment at 1K intervals within the 8080 addressing range (for example, from 3K to 11K, not just 0-8K). Card address is selected by a DIP switch, located conveniently at the top of the card. Each of the 76 Integrated Circuits has its own **premium grade, low-profile IC socket.** We've included this special feature to increase reliability, and to make assembly, testing and repairs many times easier.

8KRA 8192-word Static Memory Module **$295** (all sockets
4KRA 4096-word Static Memory Module **$154** included)

Processor Technology manufactures a full line of plug-in modules compatible with the Altair and IMSAI minicomputers. **Write us now**—we have a lot to tell you!



6200-L Hollis Street
Emeryville, CA 94608

Part 2

"Chip" Off the Olde PDP 8/E:

The Intersil IM6100

In part 1, Bob Nelson discussed the attractiveness of a PDP-8-like computer, general features of the IM6100 chip, the timing and pinouts of the IM6100 and the instruction set used by this PDP-8 compatible microcomputer. In the second installment, he continues the flow of information on this 12 bit microcomputer.

Robert Nelson
Chief Engineer
PCM Inc
180 Thorup Ln
San Ramon CA 94583

When using programmed IO transfers, one must live with a delay waiting for the usually slow peripheral device to get ready to receive or transmit data. Often this requires the CPU to traverse a "waiting loop" thousands of times between transfers. Interrupt-initiated transfers eliminate this wasted time by allowing the peripheral device to initiate each data transfer, rather than the program. Between individual transfers the CPU is free to go about its business executing the main program. This is accomplished by isolating the IO handling routines from the mainline program and using the interrupt system to ensure that these routines are entered only when an IO device-ready flag is set, indicating that the device is actually ready to perform the next data transfer.

The interrupt system allows certain external conditions to interrupt the computer program by driving the INTREQ line (pin 8) to the IM6100 low. If no higher priority requests (such as a DMA request, or control panel interrupt request) are outstanding and the interrupt system is enabled, the IM6100 grants the device interrupt at the end of the current instruction. After an interrupt has been granted, the interrupt enable flip flop in the 6100 is reset so that no more interrupts are acknowledged until the interrupt system is enabled, usually by an instruction in the interrupt handling routine.

When the 6100 grants an interrupt, it sets the INTGNT line (pin 39) to a high level, then (so it doesn't lose track of where it was in the main program, while servicing the interrupt) deposits the current state of the program counter in location 0 of main memory. This deposited address is known as the "return address." Then the 6100 fetches an instruction from octal location 0001. This is usually a JMP (direct, or indirect as required) to the start of the interrupt service routine. The last instruction in this routine must be an indirect JMP through location 0, which returns the CPU to the main program at the same point where it left at the time of the interrupt.

The 6100 does not provide, in its hardware, for nested interrupts (that is, a higher priority interrupt of a lower priority interrupt routine); but, of course, a software stack can be programmed which will provide for any degree of nesting desired.

Direct Memory Access

Interrupt transfers use the interrupt system to service several peripheral devices simultaneously, on an intermittent basis, permitting computational operations to be performed concurrently with the data IO operations. Both programmed data transfers and interrupt initiated transfers use the accumulator as a buffer, or temporary storage space, for all data transfers. Since data may be transferred only between the accumulator and the peripheral, only one 12

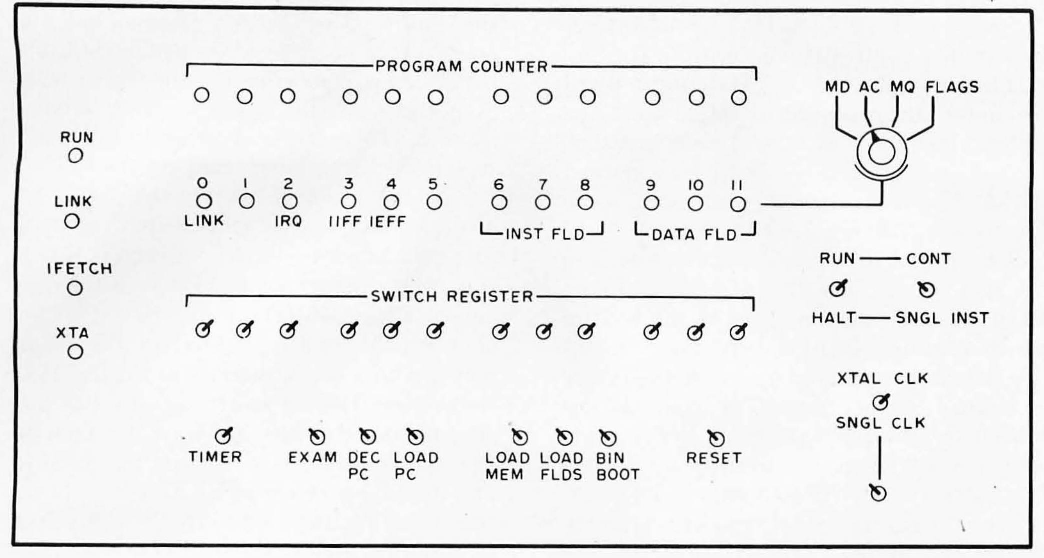


Figure 1: One example of an IM6100 control panel. This panel is designed to perform the functions of the PDP-8/E control panel. The actual logic of the control panel is performed in software of a control panel service routine contained in a special control panel memory. Bootstrap loaders and terminal monitor programs could also be implemented in the same fashion, completely transparent to the normal PDP-8/E mode of operation.

The IM6100 architecture has no hardware provisions for nested interrupts; however, a software stack can be implemented if desired.

bit word at a time may be transferred. Direct memory access (DMA), on the other hand, transfers variable size blocks of data between high speed peripherals and the memory with a minimum amount of program control required by the IM6100.

Direct memory access, sometimes also called "data break" in DEC literature, is the preferred form of data transfer for use with high speed storage devices, such as magnetic disk or tape units. The DMA mechanism transfers data directly between memory and peripheral devices. The IM6100 is involved only in setting up the transfer; the transfers take place with no processor intervention, on a cycle stealing basis. This means that main program execution will be suspended for an integral number of machine cycles while the DMA request is serviced. The DMA transfer rate is limited only by the speed of the memory and the data transfer characteristics of the peripheral device.

The device generates a DMA request when it is ready to transfer data, by asserting the DMAREQ line, pin 4, low. The IM6100 grants the request by setting a high level output on the DMAGNT line, pin 3, at the end of the current instruction. The 6100 suspends any further instruction fetches until the DMAREQ line is returned high. The data lines (DX) are put in the high impedance state, and all select lines stay high. The device which generated the DMA request must provide the address and the necessary control signals to memory to effect the data transfer.

Control Panel Features

A unique feature of the IM6100 is its provision for control panel interrupts. Due to the limited number of pins available, a microprocessor requires some sort of software implementation of its front panel controls and indicators. This is necessary because one does not have constant, or real time, access to the state of the accumulator and other internal registers and operations being performed inside. This usually requires that a portion of main memory be partitioned off for storage of an interrupt routine which updates the state of the front panel indicators or performs an operation called for by a front panel control. This routine is executed whenever a control panel interrupt is generated, as, for example, by some control switch, or an automatic timer.

Partitioning off main memory like this is, at best, an aesthetic nuisance. In the case of a machine that would handle, without modification, the software of an existing minicomputer, the partitioning becomes more of a headache since the interrupt routine may sit in some of the same memory space that the minicomputer software requires. The result may be that the interrupt routine could get inadvertently overwritten, or could interfere with operation of the mini's software.

Programmed IO transfers a word at a time; direct memory access IO sets up a special hardwired controller to take over the memory bus for transfer of blocks of data without CPU intervention.

In the IM6100 the software implementation of the control panel need not use any part of the main memory. The control panel communicates with the 6100 through the CPREQ line, pin 5. A control panel interrupt request is functionally similar to a normal device interrupt request, but with some important differences. The control panel request is granted even if the machine is in the HALT state. The CPU is forced into the RUN state for the duration of the control panel routine, then reverts back to its original state at the end of the routine. Once a control panel request is granted, the IM6100 will not recognize any device interrupt or DMA request until the control panel routine has finished execution.

When a control panel request is granted, at the end of the current instruction, the program counter is automatically stored in location 0 of a *separate* control panel memory. The 6100 then resumes operation at octal location 7777 of this separate memory. The control panel memory is organized with programmable memory in its lower pages and read only memory in the higher pages. The control panel service routine is stored in the higher pages in the non volatile ROM, with a starting address of 7777. The latter location always contains a JMP instruction which starts the actual routine at a lower address, for example, octal 7400. The programmable memory in the lower pages is used as scratchpad space.

A control panel flip flop (CNTRL FF), which is internal to the IM6100, is set when a control panel request is granted. As long as the control panel flip flop is set, CPSEL (pin 38) becomes the active select line for memory references. This distinguishes the control panel memory from main memory. However, during the execute phase of indirectly addressed AND, TAD, ISZ and DCA instructions, the MEMSEL line (pin 37) is made active. The instruction is fetched from the control panel memory, but the operand address for an indirectly addressed AND, TAD, ISZ or DCA refers first to the control panel memory for a pointer, which in turn points to a location in main memory. A main memory location may, therefore, be examined or changed under front panel control by indirectly addressed TAD and DCA instructions. Every location in the main memory is thereby accessible to the control panel routine.

At the end of the control panel interrupt routine is an indirect JMP (through control panel memory location 0), which returns CPU operation to the main program. A return address was deposited in control panel memory location 0 at the beginning of the control panel service routine, but this address may be changed by action of a front panel control operating in conjunction with a portion of the control panel service routine. Thus a "load program counter" switch might be interpreted by the control panel service routine to copy the state of an array of 12 switches (the switch register), into control panel memory location 0, which sets up the main program to start at the new address. The IM6100 provides for the inclusion of the switch register on the front panel, with a special select line. When a OSR instruction (OR the switch register with the accumulator contents, and leave the result in the accumulator) is executed, the SWSEL line (pin 31) goes low at T2. This line allows the switch register to directly drive the data lines during the "read" portion of the cycle.

The designer may also make use of the control panel features to implement bootstrap loaders in the control panel memory, so that these routines will not consume main memory space. Programs can be loaded by indirect DCA instructions: the indirect address developed in the control panel service routine points to a main memory location which is to be loaded.

Control Panel Example

It is quite easy to build a front panel for the IM6100 that provides nearly every function of the PDP-8/E control panel. For the would be constructor, one possible layout for such a panel is shown in figure 1. The software routine required to make all these controls and indicators work requires less than 128 words of 12 bit ROM. This section describes the use of software which simulates the PDP-8/E control panel functions.

The program counter display is an array of 12 LEDs, segregated into four 3 bit (octal) digits. This display shows the current state of the IM6100's internal program counter. The program counter can be loaded from the switch register by raising the LOAD PC switch, and can be decremented by raising the DEC PC switch. These operations are carried out by the control panel service routine software.

The 12 LEDs just below the program counter display are called the display lamps and show the data selected by the rotary switch to their right. With the rotary switch in the MD (memory data) position, these LEDs show the content of the memory location whose address is indicated by the program counter display. With the rotary switch in the AC, MQ or FLAGS positions, the display lamps show the state of the 6100 accumulator, MQ register or flag bits,

respectively. The flag bits displayed are indicated below their respective LEDs. These include the state of the 6100 Link flip flop (LED 0), whether an interrupt is currently being requested by an external device (LED 2), whether interrupts are being inhibited by the extended address module (LED 3), whether interrupts are enabled by the 6100 itself (LED 4), and the currently selected instruction field and data field, 3 bits each.

The EXAM switch when raised causes the data in the display lamps to be updated, and increments the 6100 program counter. This makes it possible to examine a series of sequential memory locations by loading the first address into the program counter with the LOAD PC switch. Then each time the EXAM switch is raised the address of the next memory location is displayed in the program counter; and, if the rotary switch is in the MD position, the content of that location is shown in the display lamps.

The LOAD MEM switch commands the control panel service software to load the switch register into the memory location indicated by the program counter. The LOAD MEM switch also increments the program counter after the load, so it is unnecessary to load the address for each subsequent location.

The LOAD FLDS switch is used to load the desired instruction field and data field from switch register bits 6 to 8 and 9 to 11, respectively, into the extended memory address module. If the rotary switch is in the FLAGS position, these new flag bits will be shown in the display lamps immediately after the load.

The BIN BOOT switch is used to load a binary format paper or magnetic tape (such as one produced by the DEC PAL III or MACRO-8 assemblers) into main memory. The routine for accomplishing this load can be written to fit in another 128 words of control panel service memory. Thus, with programming stored in three 1 K (256 x 4) PROMs, one can have a front panel more powerful than that on the PDP-8/E itself, since the latter does not have a built in bootstrapping capability.

The TIMER switch, when in the "up" position, turns on an oscillator which runs at about 25 Hz. This oscillator causes a control panel interrupt to be generated every 40 ms, for the purpose of updating the displays on the panel when the machine is running.

The RUN, LINK, IFETCH and XTA lamps show the state of the respective lines on the IM6100 chip.

The RUN/HALT, and CONT/SNGL INST switches are closely associated. When the RUN/HALT switch is in the HALT position, the machine is halted. In this state raising the CONT/SNGL INST switch will cause the machine to execute one (the next) instruction and again halt. This provides a very convenient single instruction operation mode for program debugging. When the RUN/HALT switch is put into the run position, the machine is enabled to enter the RUN mode. It will begin running when the CONT/SNGL INST switch is raised.

The XTAL CLK/SNGL CLK function is implemented with a pair of switches. When the upper switch is in the XTAL CLK position, the CPU runs off the internal crystal controlled clock oscillator. When this switch is in the SNGL CLK position, the machine is in the single clock mode: each time the lower switch is raised, a single clock pulse is generated. This provides a capability to step an instruction through its various phases. This capability and the single instruction mode described above are made possible by the static nature of the registers in the IM6100 chip.

Support Devices

In addition to producing the IM6100, Intersil is also offering several other CMOS devices intended to support processor and computer designs built around the 6100. These include a 1 K x 12 bit mask programmed ROM (IM6312), 256 x 1 bit RAMs (IM6523/6524), 1 K x 1 bit RAMs (IM6508/6518), a 256 x 4 bit programmable random access memory (IM6561), a parallel interface element (IM6101), and a UART (IM6402/6403).

The CMOS 1024 word ROM, since it is mask programmed, is aimed at volume production controller applications, and is not well suited to one-of-a-kind hobbyist systems. It does have a unique feature worthy of mention here, however. In many programming applications, it is very convenient to have ROM space and programmable memory space interleaved in memory, say three pages of ROM followed by a page of RAM, for example. The IM6312 contains 8 pages (1024 words) of memory and any 2 page block may be designated, in the mask programming operation at manufacture, to be dedicated to RAM space. Then when any word in this 2 page block is addressed, a RAM enable pin on the 6312 is activated which turns on RAM chips located next to the ROM. These RAM chips can share addressing space with the ROM, and an additional latch and decoder for RAM addressing are eliminated. Operation of this kind creates the illusion of a programmable area in the read only memory region of address space.

The IM6100 is designed to use a software front panel driver; provision is made for a separate 4 K word control panel memory. A periodic control panel interrupt switches from main memory to control panel memory and initiates control panel software.

The parallel interface element, IM6101, provides addressing, interrupt logic and control for interfacing the IM6100 to a variety of peripherals such as keyboards, UARTs, specialized memory, etc. It is designed to eliminate a large part of the random logic found in many device interfaces. The 6101 has sense lines that constantly monitor the status of the peripheral device to which it is interfaced. When the device indicates a ready condition, the 6101 generates an interrupt to the 6100, which initiates a data transfer. Several IM6101s may be daisy chained to form an interrupt priority hierarchy, with the highest priority devices at one end and the lowest priority devices at the other. When the 6101 generates an interrupt to the 6100, it also provides the address of the initial location in the interrupt service routine, thus creating a vectored interrupt system. The IM6101 also contains several other features too lengthy to discuss here; the manufacturer's data sheet for the chip gives complete details.

Interrupt initiated transfers eliminate timing loops.

The CMOS UART, IM6402 or IM6403, is identical to the PMOS UARTs that have been available for some time, except for some unique features. First, its CMOS construction allows it to be clocked as fast as 3.2 MHz, which is 10 times faster than the PMOS units. And of course CMOS construction means very low power consumption. Second, the IM6403 version provides an on-chip crystal oscillator and divider chain for bit rate generation. With a commonly available color television crystal at 3.5795 MHz, the baud rate becomes 109.2 Hz, which is just fine for a Teletype interface.

The CMOS RAMs are static devices, but are presently too expensive for general purpose hobby use; the 1 K x 1 bit devices at this writing still cost over $8 each. However, the day is not far off when they will cost compete with N channel RAMs like the 2102. They are superior to the N channel devices, in that they draw only milliamps at five volts when operating, and just microamps when idling. They also contain an on-chip address latch, which minimizes the necessary interface logic. A nonvolatile programmable memory can be constructed using a handful of these chips with a small on-board Nicad battery to keep them powered up when system power is removed.

Reliable rumor has it that Intersil is working on another support chip (or chips), that will contain several useful functions. The chip will contain all the logic necessary to extend the main memory from 4 K to 32 K words, eliminating nearly all the TTL logic now required to implement an extended address element module. It will also provide DMA port logic, a real time clock and a 4 K dynamic RAM interface. It may also contain the PDP-8 user flag logic, to add time sharing capability to the IM6100.

Conclusion

When contemplating the construction or purchase of a small computer for home applications, the amateur computer user should very carefully consider the question of software availability for the completed machine. Computer manufacturers typically will tell you that in any new computer design, hardware development is only about 20% of the job. The other 80% of the effort is involved in software development. This consideration makes the IM6100 ideal for the hobbyist, since a lot of the software you will need is available from an outside source already written, debugged and documented over the decade or so in which PDP-8s have been available.

The commercial version of the 6100 (IM6100C) is well within an individual's budget in single quantities. It can be purchased from any Intersil distributor. The IM6100C has a maximum clock frequency of 3.3 MHz (18% slower than the industrial version, which costs about $100), and a maximum operating temperature of 50°C, but in all other respects is identical to the more expensive versions.

The reader interested in building his own machine around the IM6100 should contact Intersil and request copies of the application notes on the chip. These describe typical circuitry that can be used to produce a full blown minicomputer very similar to the PDP-8/E. Given the past history of this market place, it should not be long before IM6100 based kits are available to ease construction of inexpensive PDP-8/E compatible machines. ■

More information about the particular control panel design discussed in this article may be obtained by writing the author at 4100A 35th Av, Oakland CA 94169 and enclosing a self-addressed stamped envelope.

Components and Parts

Instructions for this type puzzle are extremely simple; just take the words from the list and fill them into the blocks, one letter per block. The number of blocks indicate the number of letters in each word. Take your time; you may have to do a little searching to make all the words fit into the puzzle.

Robert Baker
34 White Pine Dr
Littleton MA 01460

Puzzle Time, page 84, May BYTE

ANSWER: This is but one of several possible solutions to the puzzle. However, the sum for any given row or column = 385_{10} = 601_8.

A	N	S	I	V
Y	B	L	P	J
F	X	C	O	Q
T	G	W	D	K
M	R	H	U	E

3 Letter Words
FAN
FET
LED
NUT
PLL
POT
ROM
SCR
VCO

4 Letter Words
BOLT
CHIP
COAX
COIL
CORE
DIAL
FUSE
GATE
KNOB
LAMP
PLUG
TUBE
UART

5 Letter Words
CHOKE
DIODE
METER
RELAY
SCREW
TIMER
TRIAC
ZENER

6 Letter Words
FILTER
MEMORY
PADDER
SOCKET
SWITCH
TOROID

7 Letter Words
BATTERY
CHOPPER
CRYSTAL
DISPLAY
SPEAKER
TRIMMER
VIDICON

8 Letter Words
HEATSINK
INDUCTOR
KEYBOARD
RESISTOR
RHEOSTAT
SOLENOID
TANTALUM
VARACTOR

9 Letter Words
AMPLIFIER
CAPACITOR
CONDENSER
CONNECTOR
INSULATOR
PHOTOCELL
RECTIFIER
REGULATOR
THYRISTOR

10 Letter Words
ATTENUATOR
THERMISTOR
TRANSISTOR

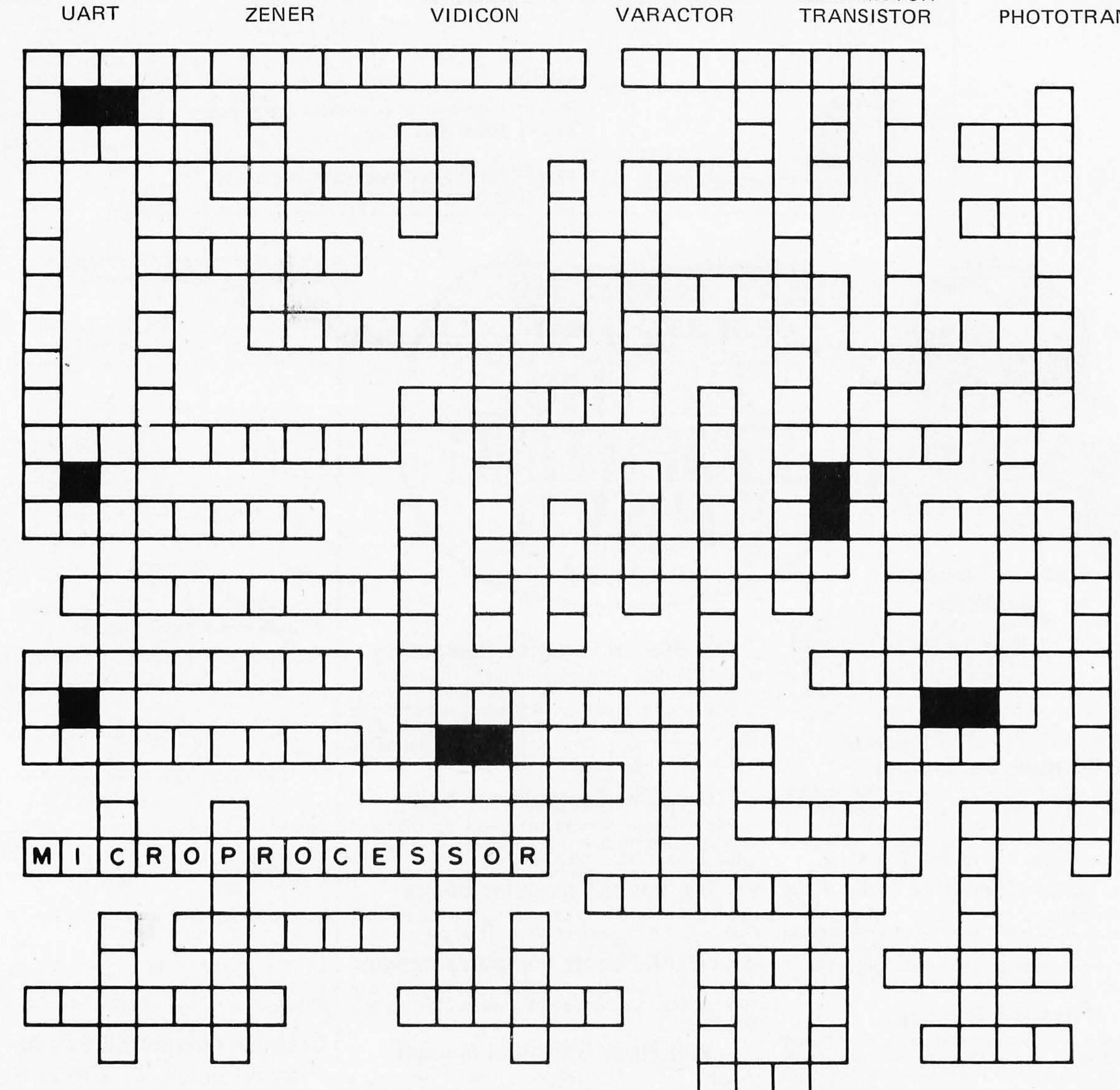
11 Letter Words
TRANSFORMER
UNIJUNCTION

12 Letter Words
ELECTROLYTIC

13 Letter Words
POTENTIOMETER

14 Letter Words
MICROPROCESSOR

15 Letter Words
PHOTOTRANSISTOR



HELP WANTED (A BYTE OPPORTUNITY)

BYTE needs a person to assist with technical editing of articles and books. Applicants should have a strong background in hardware that is evolving toward an understanding of software, should be literate and capable of constructively criticizing manuscripts and composing copy when needed, and be willing to work hard and meet deadlines consistently.

BYTE offers a congenial working environment, a lovely small town New England location, complete medical benefits, profit sharing and fun.

Interested parties should respond in writing (do NOT telephone) with a complete resume including salary requirements. Please enclose a short essay giving your reasons for wanting to work for BYTE. We do not expect to be able to respond to all applications individually but will return all resumes if an SASE is supplied. Address applications to: BYTE, Resumes Editor, 70 Main St, Peterborough NH 03458.

WANTED

Microprocessors Kit or Assembled

ALTAIR • SPHERE • MOTOROLA
INTEL & OTHERS

Memories, I/O boards, chassis, power supplies, modems, peripherals, specials, prom programmers, ADCS, DACS, GPIOS, RS232 faces, and any associated software.

FORWARD INFORMATION AND/OR INQUIRIES TO:

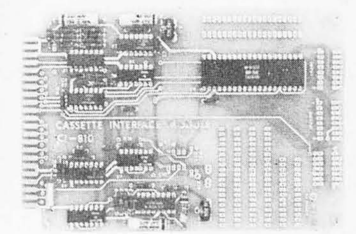
AMERICAN USED MICROPROCESSORS, EQUIPMENT & SUPPLY CORP.
BOX 515,
PRAIRIEVIEW, ILLINOIS 60069
TELEPHONE (312) 634-0076

For complete monthly catalog of items available and in stock Enclose $10. for six issues. Many new unused items. Amazing values for everyone. A-70

TIRED OF LOADING YOUR COMPUTER BY HAND?

TURN AN AUDIO CASSETTE RECORDER INTO A FAST RELIABLE PROGRAM LOADER AND MASS DATA STORAGE UNIT WITH THE *PerCom* CASSETTE INTERFACE FOR $59.95

- Fully assembled and tested
- BYTE Standard (see Feb '76 BYTE)
- Load a 1K program in 40 seconds
- May be upgraded to 218 bytes/sec
- Store 100K bytes on C-60 cassette
- Standard 8-bit parallel interface
- Easily connected to any computer
- Little or no software required
- Provision for custom interfaces
- Signal level sensor
- Sidetone circuit
- Operate 2 tape units simultaneously
- Standard sized circuit card (4½" x 6½")
- Described in BYTE (March-April '76)
- Comprehensive Instruction Manual



TO ORDER: Send check or money order for $59.95. Texas residents add 5% MasterCharge, BankAmericard welcome, include all numbers and sign your order. Allow 2-3 weeks for delivery, personal checks take longer Full refund if not satisfied.

PerCom – 'peripherals for personal computing'

Custom Design Services • 4021 Windsor
Garland, Texas 75042 • (214) 276-1968

MICROCOMPUTER HARDWARE

—featuring MITS Altair—

1: **HARD DISC UNITS**
(from 5-8 megabytes to 300 megabytes)

2: **ALTAIR CONTROLLER**
(For our disk units, or your own)

3: **16K PROM CARDS**
(Ask about our operating system on a card)

THE COMPUTER SYSTEMCENTER

3330 Piedmont Road
ATLANTA GA 30305
404–231-1691

Computer Mart of New York Inc.

Authorized dealer for:
Sphere • IMSAI
Processor Technology
National Semiconductor Scamp

Featuring a complete line of technical publications and books and a full line of chips and parts at **MAIL ORDER PRICES.** Repair service for all 8080 and 6800 microprocessors

Friendly Service Advice
Problem Solving

Open Monday through Saturday 10-6, Thursday until 9:30

314 Fifth Avenue
New York NY 10001
212 279-1048

MICROCOMPUTER SOFTWARE

—featuring MITS Altair—

1: **ACCOUNTING**
(General ledger; payroll; receivables; payables)

2: **INVENTORY**
(On-line control; point-of-sale optional)

3: **WORD PROCESSING**
(Create and edit via video display)

THE COMPUTER SYSTEMCENTER

3330 Piedmont Road
ATLANTA GA 30305
404–231-1691

Interact with an ELM

G H Gable
419 Jackson St
Grand Ledge MI 48837

The fundamental interface between the user and the hardware of a computer system is the system software. It runs the gamut from a dozen or so bytes of a bootstrap loader on a microcomputer to the multimillion word operating system of a large general purpose computer system. In fact, the microcomputer system can be made to do much of what the general purpose computer does with appropriate versions of systems software. One of the most significant differences between the big computer and the microcomputer is that the large computers typically operate on multiple bytes of information and often provide extended arithmetic and logical operations. Minicomputers and microcomputers can emulate these extended operations with software; the main difference is speed. The typical large computer might execute its built in instructions 1000 times faster than a microcomputer's software emulation. However, all the features of a large computer system can be implemented in the software of a microcomputer system. This includes assemblers, compilers, text editors, timesharing and multiprogramming, disk operating systems, virtual memory, utilities, and of course applications programs. In addition, the powerful hardware of a big computer can be emulated with software. The principal hardware requirements, other than a general purpose instruction set, are access to the program counter, an interrupt structure and possibly direct memory access by the peripheral equipment. Program counter access and interrupt processing is available in most microprocessors; direct memory access is often implemented by peripheral device controllers using the system bus.

For microcomputers, the system software can be divided into two major segments: the operating system or monitor and a utility library of functions which extend the instruction set. The utility library is a set of subroutines written to redefine and expand the operations the computer can perform. It can range from a simple set of number conversion and formatting routines up to the complexity of a complete floating point mathematical package.

Monitors

The monitor program, sometimes called the executive program or operating system, is the program which the computer executes when it is not running some other program. The monitor's primary purpose is to decide what the system is to do next. Sophisticated monitors typically implement disk operating systems, time sharing and multiprogramming. They call loaders, assemblers and compilers, handle input and output, and process user requests. In short, the monitor program is "the brains" of the system. In some very large systems, such as the Control Data Corporation's CDC-6500, the monitor program even has its own processor, separate from the central processors. The central processors are merely slaves to the monitor processor in such a multiprocessor system.

For a beginning, let's examine a very simple monitor program. If you have a microcomputer which needs system soft-

ware, this might be just the ticket to get you on the system. This monitor design will let you load and execute programs and edit the contents of memory. From such a basic monitor, more sophisticated software can be developed to upgrade the system to whatever level you desire.

ELM

Every routine should have a name, especially a system routine. Thus I call this the Eloquent Little Monitor, or ELM. ELM is designed to have a Teletype or a cathode ray tube (CRT) terminal such as a TV typewriter as its control console. A CRT running at 1200 baud makes a wonderful control console due to the brisk speed at which messages are transmitted. ELM implements four commands in its simplest version: LOAD which will load a program into memory beginning at a specified location; LIST which lists the content of selected memory locations; MODIFY which will modify the contents of selected memory locations, and GO which starts execution of a program at a specified location. My version of ELM features decimal addresses and allows input line editing.

Many processors begin execution at a fixed location at power-on or system reset. Some processors begin execution at a hardware programmed address which might be set by switches. Wherever the processor begins its execution, the implementation of ELM assumes that ELM will be the program which receives control as a matter of course. For the purposes of this article, we'll assume that ELM is located at the low end of memory address space. Following ELM comes the first available address (FAA) of user programmable memory, then the last available address (LAA). This memory organization for a typical monitor residing at the low end of address space is shown in figure 1. Other allocation schemes are of course possible. It is also desirable to have the monitor in a read only memory so that, when the computer is first switched on, the CPU will immediately begin execution of the monitor. With such a firmware monitor, your programs will not be able to destroy the monitor program itself. In addition to the address space for the monitor, the allocation shown in figure 1 includes 80 bytes of programmable RAM for use as data storage.

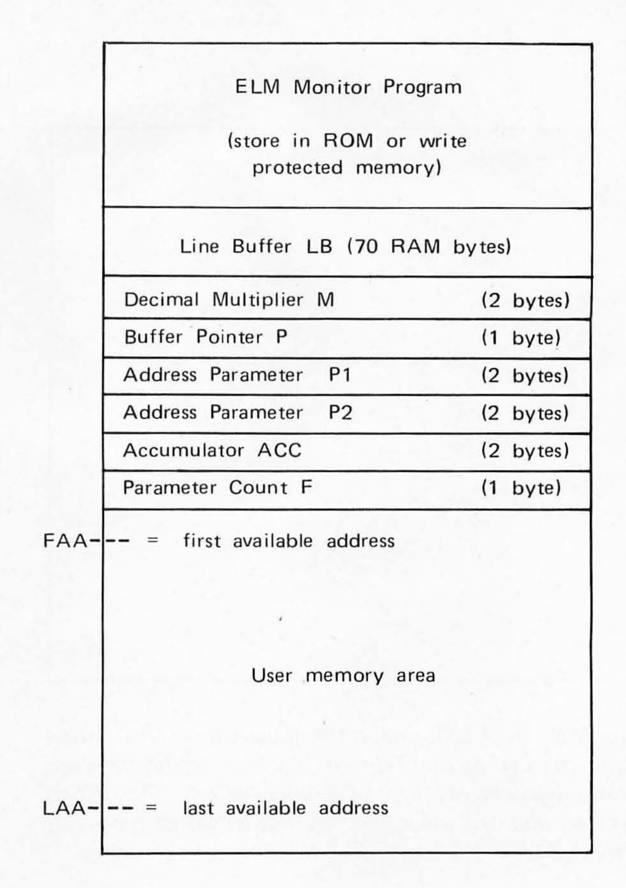


Figure 1: Memory Allocations for a Typical Monitor Program. This map assumes that the ELM monitor program resides at the low end of memory address space, and that programmable random access memory begins at the address of the line buffer.

Using ELM

First, let's look at the monitor from the user's point of view at the terminal. When the system is switched on, the Teletype or display will print "OK–". Any time the monitor is waiting for a command it will print the same message.

If you want to enter a program starting at location 123, type "LOAD,123" then a carriage return to end the line. The ELM program will respond with the message "123=" on the next line. ELM now expects you to type a string of hexadecimal characters grouped two per byte, with a single space between each group. See figure 2 for examples of this format. You can enter up to 22 double character hexadecimal codes on a line. The line is terminated with a carriage return. After the carriage return, these codes are entered into memory beginning at the address 123 in this example. Then the address waiting to be loaded will be printed at the beginning of the next line so that more hexadecimal codes can be entered. This process is repeated until you type the word "END" at the beginning of a line. After ending the load routine, the last loaded address is printed followed by the "OK–" message which indicates that ELM is

```
ØK- LØAD,1104
1104= 12 7E 51 C3 69 01 63 5A
1112= 04 5C 54 12 43←8
1117= 49 2C42 59 54
ERRØR
1119= 42 59 54 53 20 2C
1126= END
LAST ADDRESS LØADED 1128=

ØK- LIST,1104-1128
11C4= 12 7E 51 C3 69 01 63 5A 04 5C 54 12 48 49 20 42 59 54 45 53
1124= 20 2C 5A 00 00

ØK\\MØDIFCY,11
MØDIFY,1112
1112= C4
1112= 06

ØK- LIST,1111-1113
1111= 5A C6 5C

ØK- GØ,1024←←←1C4

HI BYTERS

ØK-
```

Figure 2: Sample Printout of an ELM Interactive Sequence. This listing shows ELM at work. Note the use of the Teletype back arrow (underscore character) to delete mistakes and one instance of a cancelled line. This listing illustrates use of ELM to load and execute a simple program which types out "HI BYTERS" and returns to ELM.

back in the command mode again. If the starting address is omitted or is less than the first available address (FAA) then FAA is assumed.

If you want to list the contents of memory locations 123 to 456, the command "LIST,123-456" will start the listing, printing 20 hexadecimal codes per line. If the address range is omitted, listing begins at the first available address (FAA) and continues until the last available address (LAA) or an end of program mark. Figure 2 illustrates the output format of a listing.

If you want to modify memory contents at locations 123 to 130, the command "MODIFY, 123–130" will first list the old contents of these locations, then it will enter the load routine to print "123=" as if you were loading these locations. Modified codes may then be entered, to be stored beginning at 123.

Finally, if you want to start executing the program at location 123, the command "GO,123" puts 123 into the program counter and begins execution of your program. Again, if the address is omitted, execution starts at the first available address, FAA.

It is certainly easy to make typing errors, especially for me. Thus I implemented ELM with a line buffer and two special line editing characters. The underscore (ASCII back arrow, hexadecimal 5F) effectively removes the preceding character typed, two underscores remove the preceding two characters, etc. The control X character (ASCII cancel code, hexadecimal 18) cancels the whole line. Several reverse slashes (ASCII, hexadecimal 5C) are printed on the cancelled line and a line feed is generated as shown in figure 2.

Architecture

Now that the monitor design is set, let's look at the architecture of the program needed to implement ELM. Figure 3 shows the logic for the whole monitor. After the power on restart, "OK–" is printed as the ELM input request message, then the system idles while waiting for input. Figure 4 shows the logic of the subroutine INPUT, which reads each character and puts it into the line buffer. If the terminal is running in the full duplex mode, the character should be echoed back to the printer. The buffer pointer, P, shows where to put the next character in the buffer. The editing characters are implemented as shown. An ASCII carriage return code (hexadecimal 0D) ends the input sequence. The test for carriage return is done after storing the input character since the load routine expects a carriage return as an end of line character.

In figure 3, the parameter decoding and error checking logic is shown as a box and an error test with a note attached. This logic is expanded in more detail in figure 5. The parameter decoding logic has a structure that enforces a non ambiguous syntax on the command line. The command is examined by means of a command list. This list is a table which is sequentially searched, matching the command in the buffer with each possible command in the table. The result is used to determine the proper branch. An error message is printed if the command is not found in the table.

The LOAD subroutine is shown in figure 6. The logic consists of an outer loop for each line of input, and an inner loop which scans the line, loading memory from left to right in ascending address order. The LOAD routine checks the syntax for double character hexadecimal codes separated by blanks. If a syntax error is found, loading stops, an error message is printed, and the next address to be loaded is printed on the next line. A variable number of hexadecimal codes from 1 to 22 may be entered on each line. The initial address (P1) is incremented during the loading routine.

Note that after loading is completed and control returns to the main routine, an end of program mark is inserted into memory. In my version of ELM, the code for a jump to

address zero is loaded into the next three bytes as an end of program mark. This convention allows normal termination of a user program by running off the end and branching to the starting address of the monitor at location 0.

The LIST routine is shown in figure 7. This routine simply prints out the hexadecimal codes found at locations specified by the input parameters. This listing is done 20 bytes per line. Note that LIST stores the

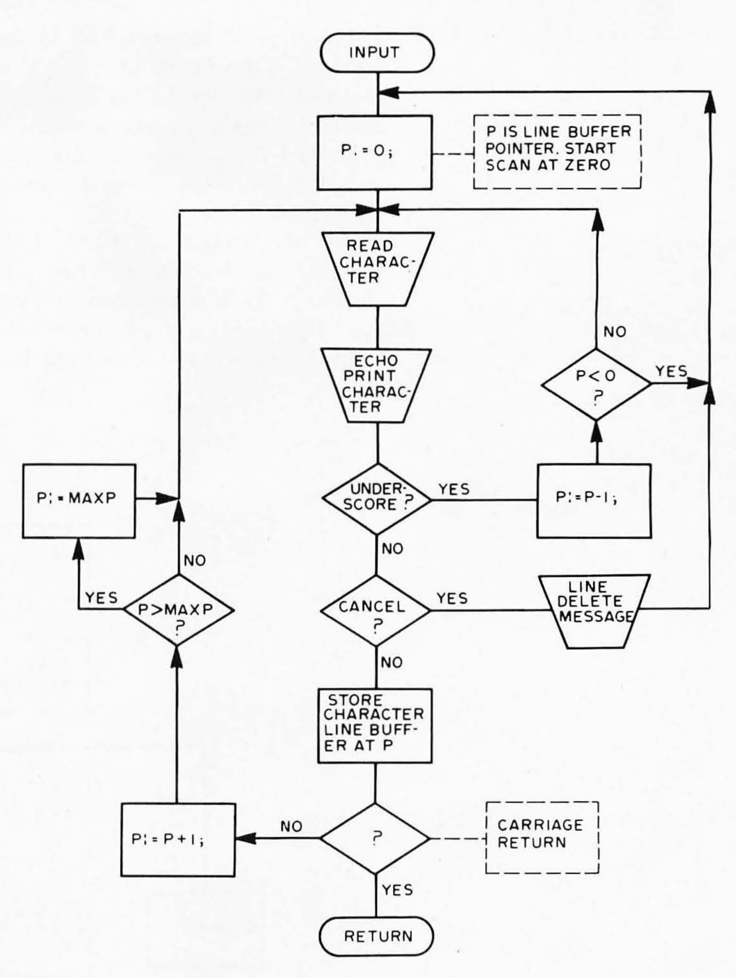


Figure 4: The Input Subroutine Specified as a Flow Chart. The main purpose of INPUT is to read one line of input, terminated by a carriage return. INPUT implements the line editing functions of character delete and line delete. When the carriage return code is detected, the line buffer LB is filled from position 0 to position P.

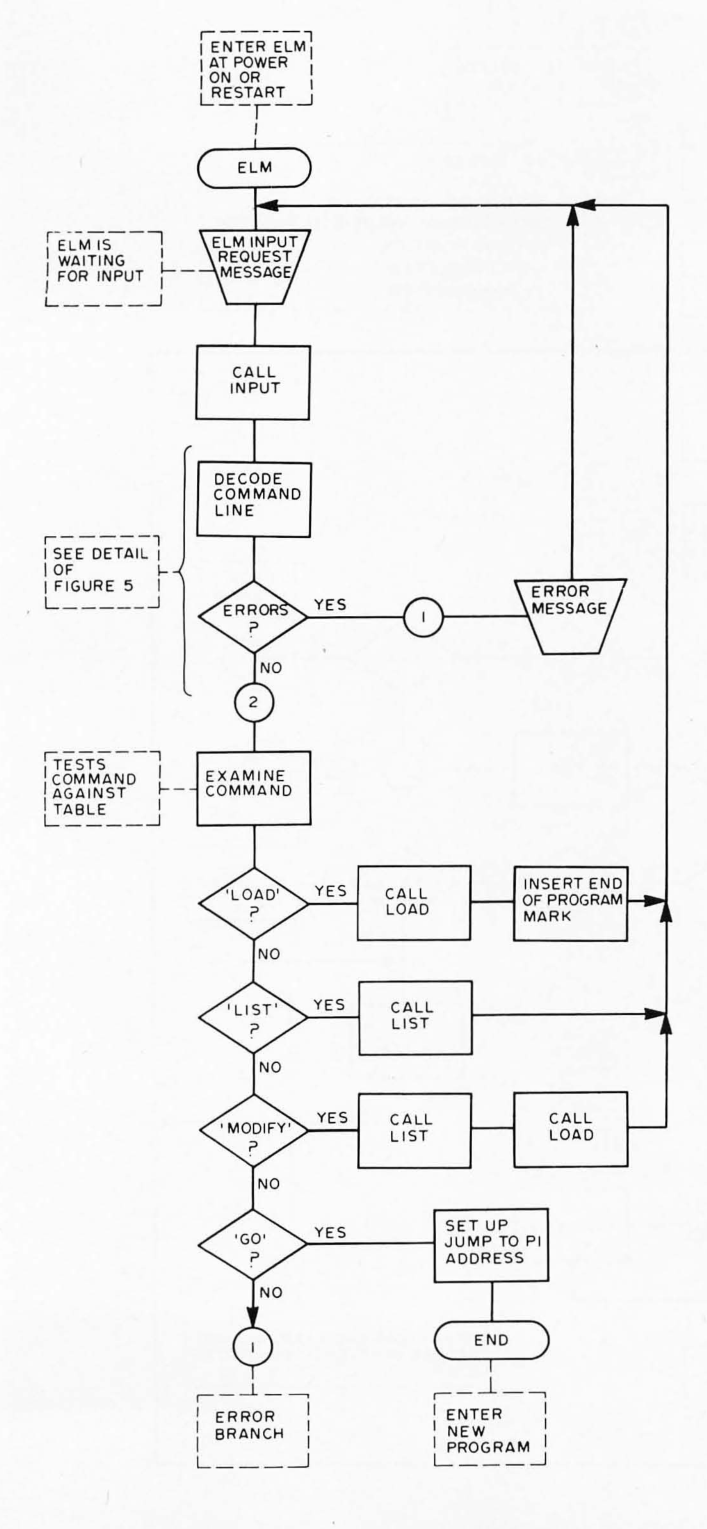


Figure 3: The ELM Program Specified as a Flow Chart. The main logic of the Eloquent Little Monitor is shown in this diagram. Flow begins at the top left and proceeds down the diagram. Normal operation of ELM involves a closed loop, returning to the ELM input request message printed near the top of the diagram. If the GO command is carried out, execution leaves ELM and proceeds to the selected address.

initial value of parameter P1 in the accumulator ACC during its operation. Then P1 is restored after the listing is completed. This allows LOAD to be called after LIST during a MODIFY sequence, so that both LOAD and LIST reference the same starting address.

In my version of ELM, addresses are handled as decimal numbers. This is reflected in the input numeric conversion logic (see figure 5) and in the creation of an output conversion subroutine: Both LOAD and LIST call a subroutine *DECIMAL which* prints the decimal addresses at the beginning of lines in messages. DECIMAL simply converts the first address parameter, P1, into five ASCII numeric characters, and prints them followed by an ASCII "=" character and a blank. I put decimal address conversion into ELM out of personal preference. The decimal conversions may be omitted and hexadecimal or octal address parameters could also be used. There is already a binary to ASCII hexadecimal routine implicit in the

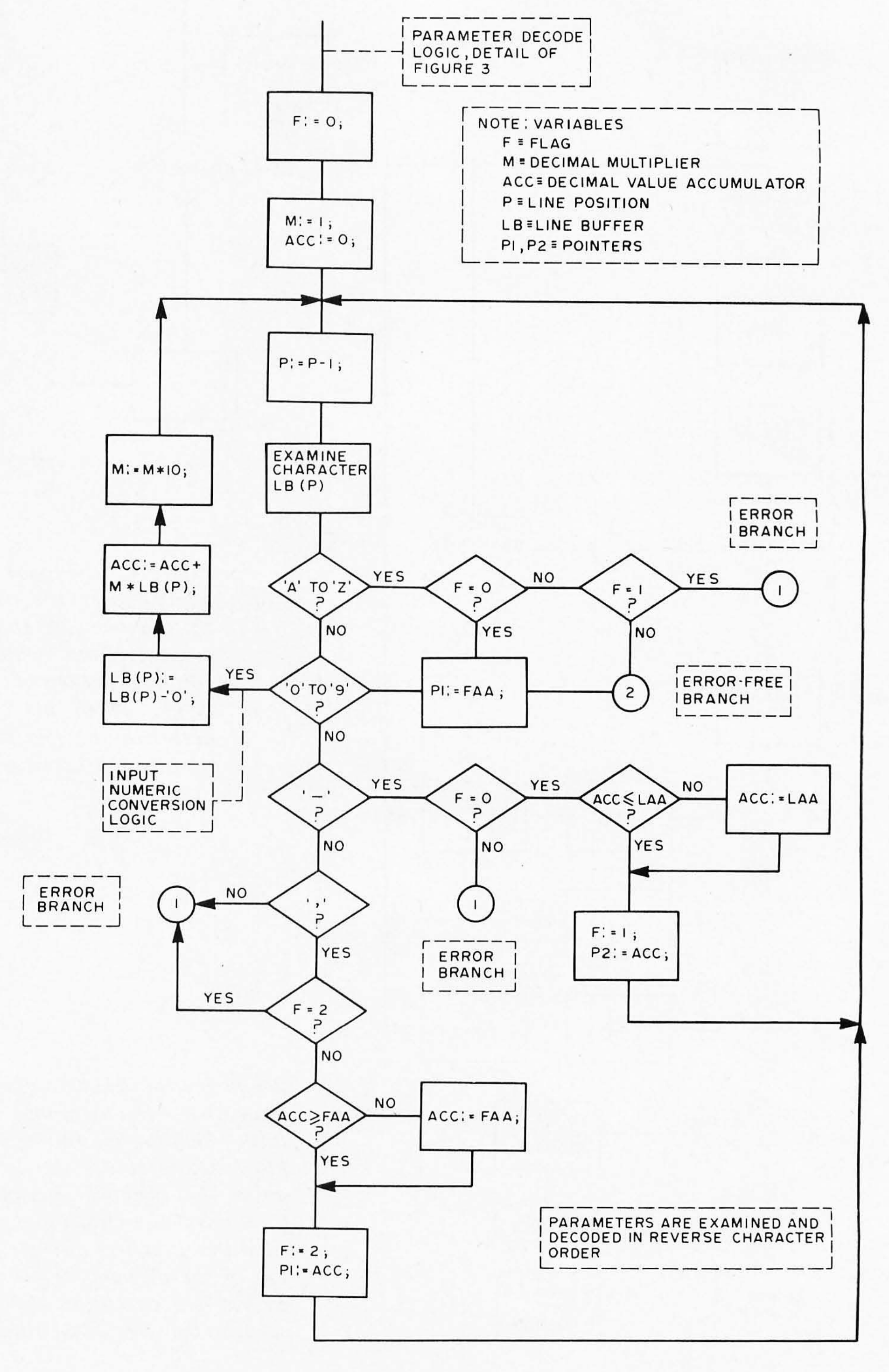


Figure 5: Parameter Decoding Logic Details. Figure 3 contains a box labelled Decode Command Line and a conditional test labelled Errors, with a note referencing figure 5. This figure contains the details of the logic needed to decode a command line into two parameters and a command. There are two possible exits from this logic. An error exit to terminal (1) occurs if an error is detected; an error free exit to terminal (2) occurs if no errors are detected.

LIST function. For input, the parameter decoding routine can be simplified somewhat by using hexadecimal parameters.

Expansions

There are several obvious expansions to ELM which should be easy to implement. You may even want to incorporate them into your own version of ELM right from the start. If you have an ASR Teletype (with paper tape reader and punch), you may want to add the following commands: LOADPT and PUNCH. Your Teletype should be able to receive the rubout character (ASCII delete, hexadecimal FF) but not transmit as is the normal configuration. LOADPT would operate the same way as LOAD except that there is no printing needed. The format of the tape would be lines of hexadecimal codes with a carriage return and two or more delete characters at the end of each line. You can skip the blanks between bytes to save tape if you like. When the processor sees the carriage return, it begins loading memory from the line buffer. The two delete characters give the computer time to load the line, so that by the time the next real character comes along the computer is ready for it. Instead of the word "END" at the end of the input, you might want to use

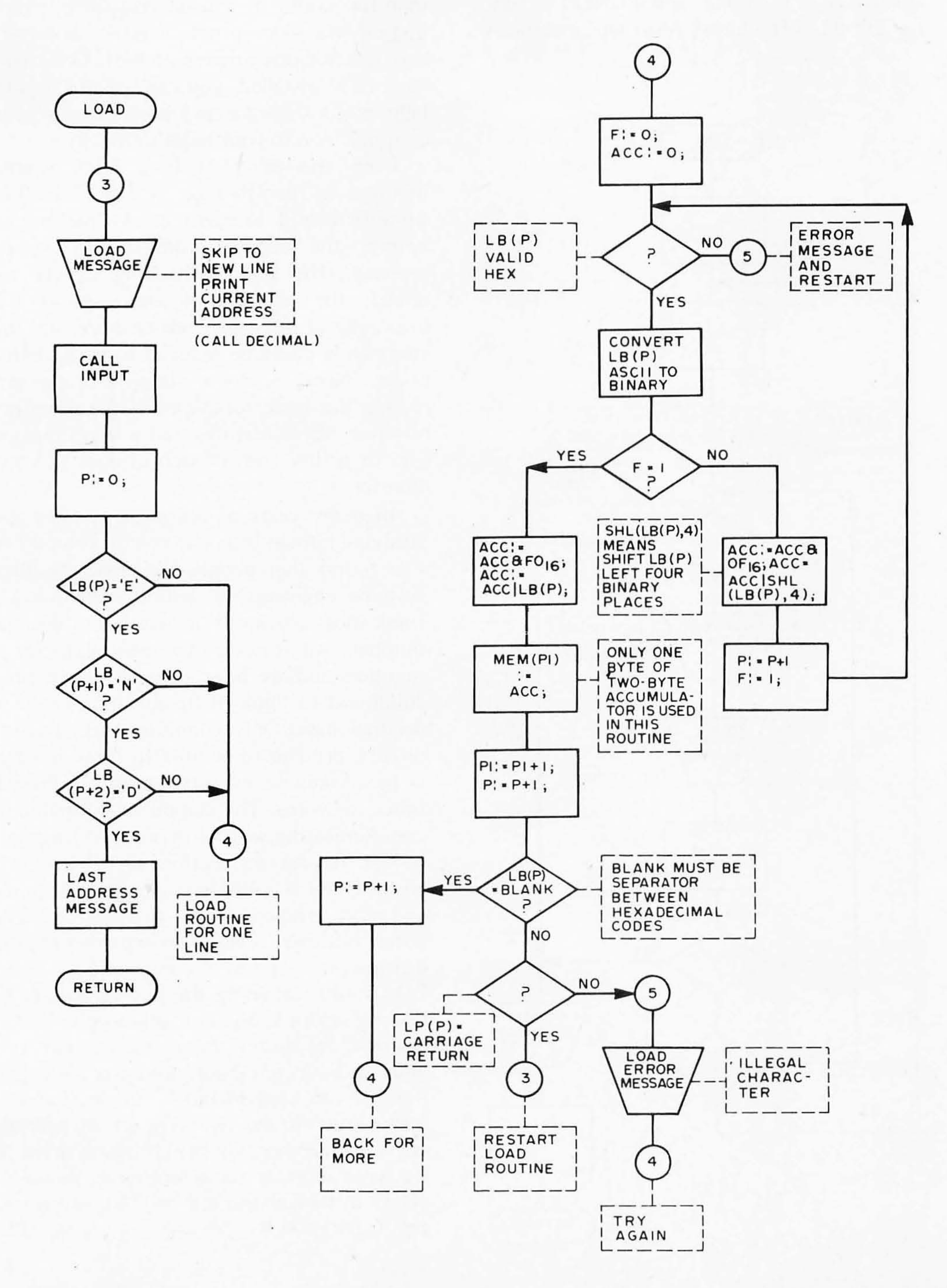


Figure 6: The LOAD Subroutine Specified as a Flow Chart. The purpose of LOAD is to set the contents of user programmable memory beginning at a location specified by the user. The routine continues indefinitely until the characters "END" begin a line of input.

the ASCII end of tape character (hexadecimal 04). The PUNCH routine would operate like LIST, without the addresses. It should punch the tape in exactly the same format read by LOADPT. If you are not using blanks between bytes in the tape format, you can get 34 hexadecimal codes on a line followed by a carriage return and the two delete characters. The last character punched might be the end of tape code or the END convention, depending upon your own preferences.

If you have a serial tape drive at a different IO port, you may want LOADMT and SAVEMT commands. These could be exactly like LOADPT and PUNCH except for the IO port address. Most tape interfaces are set up to use the *null code* (*hexadecimal* 00) instead of the delete code to give blank spacing. You may also want to implement absolute binary versions of SAVEMT *and* LOADMT to allow higher speed and eliminate conversions.

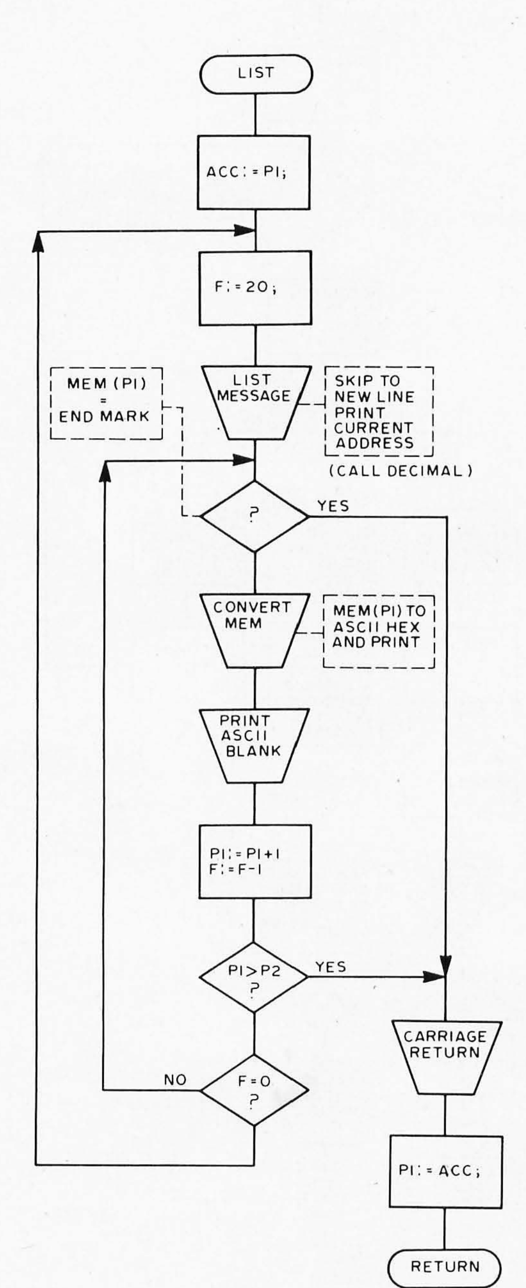


Figure 7: The LIST Subroutine Specified as a Flow Chart. The purpose of LIST is to dump the contents of memory, formatted as ASCII encoded hexadecimal digits. The dump routine types the address first on each line, then follows with 20 groups of two hexadecimal digits.

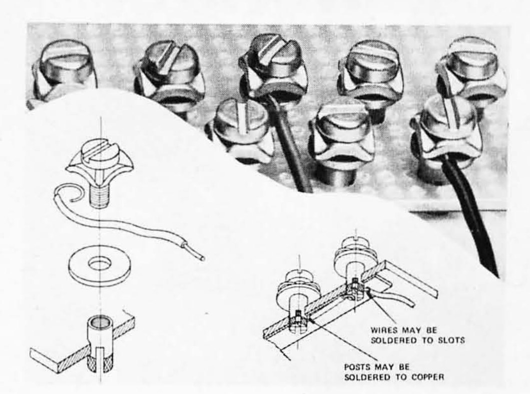
Philosophy

With this article, I've given you enough information on the design of a monitor to enable you to write the code for your own machine. After a few days of coding and debugging, you should be ready to go to the local computer store and have your ROMs zapped with a mighty ELM. The whole monitor could be put in and initially debugged via front panel switches; however, this is a tedious process at best. Once you have ELM installed, you can use this tool to help build software and programs on your own machine to your heart's content.

Even though ELM is a fairly simple monitor as monitors go, it can be further simplified and condensed. As mentioned before, the decimal conversions can be omitted. The syntax checking can be reduced, the printing of addresses at the beginning of lines might be omitted, and the commands could be reduced to single letter codes. None of these simplifications will reduce the basic functions of the monitor; however, these features add a sharp dimension of utility and a touch of class to your monitor.

In many years of designing systems and studying human interaction with computers, I've found that people (ie: users, be they systems engineers or airline ticket clerks) think most efficiently in words and decimal numbers. Addresses are a sequential stream of numbers and we have all been taught since childhood to think of streams of numbers in decimal base. Only computer nuts, putting on airs, pretend to be able to think in octal or hex. Likewise, we communicate with each other in words. The computer is capable of communicating with us in our own language, so let it. An instruction such as LOAD STARTING AT 489 is much easier to learn and more efficiently used than L,01E9. The latter, however, is easier to implement in the computer. ELM compromises with LOAD,489; retaining the keyword and the decimal address. My basic philosophy is: *Let the machine do the things it is good at.* It is good at base conversions and word recognition. It can convert binary to decimal and back again in the twinkling of an eye; we can't. Remember, you will probably want to use your monitor for a long time; the extra effort in its construction will be well worth the frustration.■

What's New?



Vector's Done It Again. . .

Here is another neat little innovation from the people at Vector Electronic Company, Inc: a screw down terminal post intended for prototype applications of Vectorboard. It's fairly obvious that the model T122 terminal will also find its way into kit products, since all one would have to do is solder it into an appropriate pad hole in a printed circuit board in order to

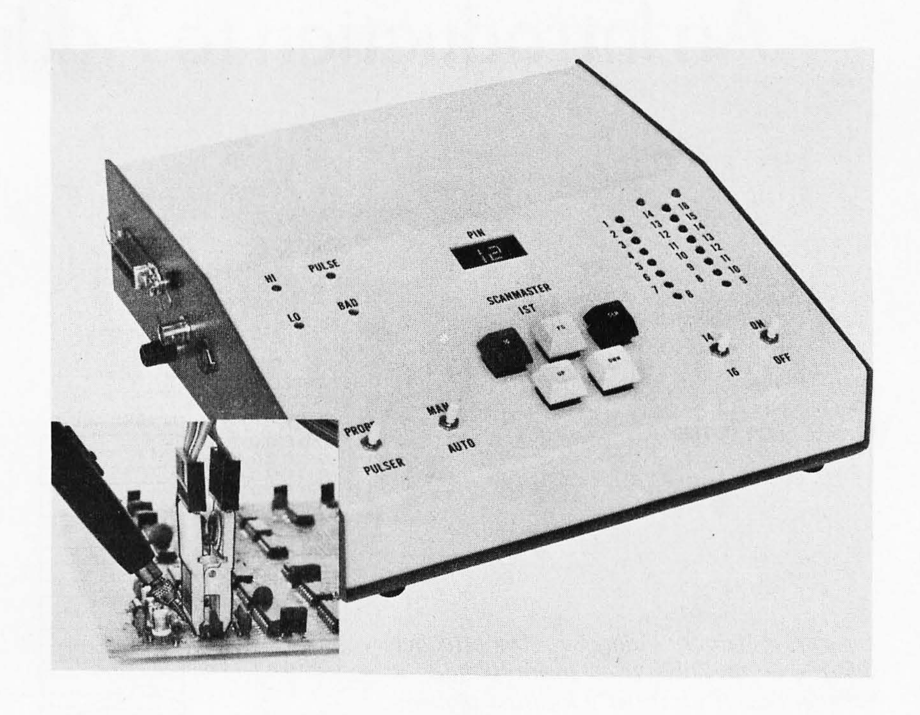
obtain considerable flexibility in wiring up such household items as burglar alarms, thermostat sensors, fire sensors, model railroad layouts, etc, which might be controlled by the interfaces of your computer. The price for individual users is $1.40 per package of 10, $12 per package of 100 terminals. Contact Vector Electronic Company, 12460 Gladstone Av, Sylmar CA 91342.■

Short Courses

Massachusetts Institute of Technology is offering two short courses during its summer session, courses which might prove useful to readers with a professional interest in computer science. Scheduled for June 7-18 is "Mini-Micro Computer Systems: A Close Look at Minicomputers/Microcomputers and Their Applications to Real World Problems." Scheduled for August 16-27 is a course entitled "Operating Systems." For detailed information, write the Director of the Summer Session, MIT, Cambridge MA 02139.■

New Technology Breeds New Test and Service Equipment

The new technology of microcomputers brings with it the demand for new and improved circuit testing mechanisms. For the people who operate microcomputer repair facilities, this Information Scan Technology Scanmaster Logic Tester, Model 5700A, may prove a quite useful adjunct to the usual scope, digital volt meter and counter testing equipment. What this unit accomplishes is a prescan of TTL logic levels on the 14 or 16 pin circuit being tested. In a repair situation, this device can be clipped to each TTL IC in turn to automatically (or manually) test the logic levels of each pin against the TTL specification: Any pin with a voltage level between 0.6 V and 2.6 V under static conditions will be measured and flagged as a TTL level error which can be read out by pin number in the unit's display. Any dynamic (changing) signal which does not meet the TTL threshold specification will also be flagged (provided its frequency is under 10 KHz.) The threshold levels can be adjusted by the user if desired. When the



Scanmaster finds a bad level, it stops its scan and displays the number of the erroneous pin. Checking can be resumed by pressing an appropriate ignore (IGN) button. For detailed analysis, a logic probe can be used for pin by pin tests. The unit is priced at $895 FOB Santa Clara CA and comes with a standard 16 pin test cable and clip. Contact Information Scan Technology, 3650 Charles St, Santa Clara CA 95050, (408) 988-2789.■

An Introduction to Addressing Methods

John Zarrella
90-9 Wakelee Rd
Waterbury CT 06705

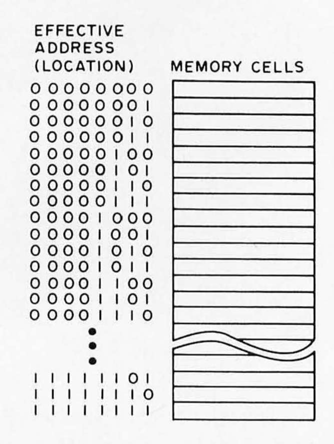


Figure 1: Memory Addresses. The effective address is the object of memory address calculations. It identifies a location in memory address space for the particular cell involved in some operation.

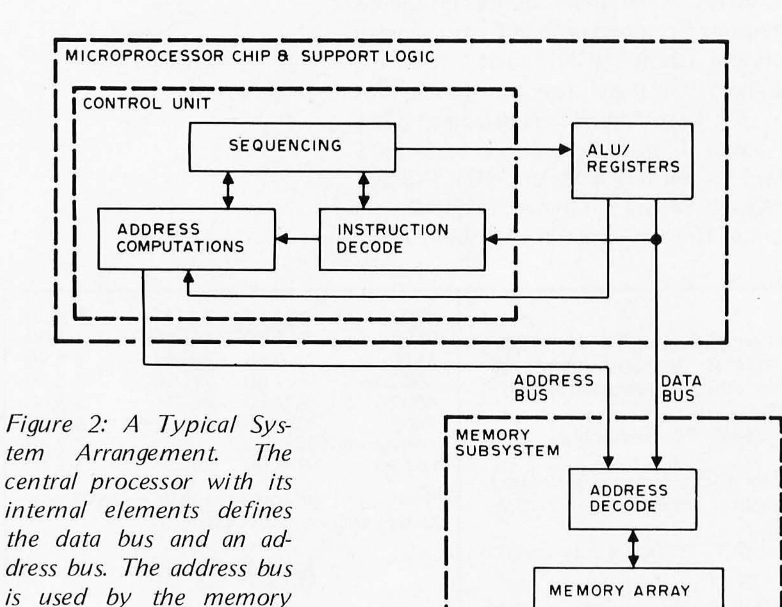


Figure 2: A Typical System Arrangement. The central processor with its internal elements defines the data bus and an address bus. The address bus is used by the memory subsystem to decode a particular location in the memory array which will be connected to the data bus.

An address is an identifier which describes the location of a particular piece of information within a computer's memory system. This information, when presented to the central processing unit for use in a computation, is usually referred to as an operand. In all microprocessor systems and in most other computer systems, an address is a binary number which is decoded to reference one computer word of information somewhere in the memory subsystem. Figure 1 illustrates how unique addresses are typically associated with memory cells.

It is interesting to note that this identifier need not be a number. There are some experimental computer systems in which memory locations are actually referenced by name or a combination of a name and a numeric index during execution. In these systems, there is hardware which translates the name directly into the location of an appropriate memory cell or group of cells.

In a similar manner, when writing programs in either assembly language or a higher level language such as FORTRAN, a programmer uses names to reference information. In this case, however, the names are generally mapped into numeric addresses by the language processing program and are not actually implemented in hardware as named references.

Instruction Cycles

Figure 2 illustrates typical interconnections among the control unit, arithmetic and logic unit (ALU), registers and memory subsystems of a general purpose processor. A brief review of the typical instruction fetch and execute cycle of such a CPU will be useful for the discussion which follows. The instruction fetch begins when the control

unit requests the next instruction by transmitting its address to the memory subsystem via the address bus. The current instruction address is usually maintained in a register called the program counter (or PC), and is updated to point to the next instruction when the current instruction is completed.

The information returned is treated as an instruction which specifies what function is to be performed by the processor. This instruction is analyzed in the instruction decode section of the processor. The execute portion of the instruction cycle then performs the functions which are specified by the decoded instruction.

Most instructions require data operands from the memory subsystem before execution can be completed. Thus a memory address must be created and sent to memory. This address is created using information contained in the decoded instruction in conjunction with information contained in various registers of the processor. The process of determining a data address is called address formation or address computation and is performed by the address computation section of the central processor. The result of address calculation is called an effective address.

A number of address formation capabilities are provided in the various designs of computers which are available. The typical contemporary microprocessor only provides a portion of the address calculation options to be described below. However, each mode, when available, can be utilized advantageously by the programmer. An understanding of addressing modes is useful when evaluating the instruction set of a computer. In order to clearly define the variety of addressing methods, an analogy will be used in the following discussion.

Immediate Addressing

In many ways memory addressing may be likened to the postal system. Imagine that you are writing a book on atomic physics and that Dr J Smith is to be a consultant. He currently lives in a small apartment complex called Apple Valley at 15 Grove St. There are five apartments at this location, each of which has its own street number—from 15 (manager) to 19. The mailboxes are arranged as shown in figure 3.

While researching the book, you attempt many of the necessary calculations yourself. These calculations involve multiplication, addition, transcendental functions and so on. Many times in these calculations you use fixed numeric factors, such as 18, which approximates $2\pi^2$. In doing this, you are treating 18 as a simple integer constant for the purposes of the approximation. In computer addressing terminology, this constant might be referenced with what is called immediate addressing by simply putting the number in a field of the computer instruction which follows the operation code. Here the effective address of the data is derived from the current program counter, and the actual instruction contains no addressing information.

Direct Addressing

Many times when performing calculations, you find that the results obtained are perplexing and need explanation. Therefore, you decide to ask your consultant for help. Since Dr Smith does not believe in telephones, you must send him a note, addressed to:

Dr J Smith
18 Grove St

In this case, the value of 18 is being used as an address. When delivering the letter, the mailman uses this address to determine where the letter belongs on Grove St. In its computer form, addressing with a single number such as 18 is called direct addressing or absolute addressing. In a computer, this number forms the address field which follows the instruction code in the program. This address field contains all the information needed by the memory subsystem in order to reference the required information, in the same manner that 18 Grove St contains all the information needed to locate Dr Smith on Grove St.

Note the contrast of this use of 18 as an address with its previous use as a constant. The number 18 which follows the instruction code is the same in either case; the intended use differs according to the instruction being executed. To know whether to use a number following the instruction code as an address or as a constant, its context must be known. In the typical computer, this is accomplished by building a special set of instructions called immediate instructions which use the number following the instruction code as a constant. A second set of instruction codes will be devoted to the absolute addressing mode, in which the field following the instruction code is an address. In general, for each possible addressing mode, a set of instructions exists which uses

An effective address is the goal of address calculation techniques.

The problem of computing a result often reduces to the problem of organizing the reference of operands in memory through addressing techniques.

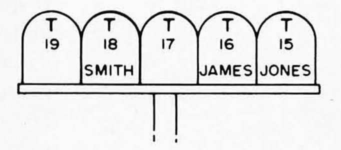


Figure 3: The concept of a memory address can be likened to that of a post office address.

that mode and interprets the information following the instruction code according to that mode.

Addressing With Registers

Suppose that you did not know Dr Smith's street address and sent the letter anyway. When the letter is received at the post office, the postmaster, knowing Dr Smith very well, would have to tell the postman: "I can't remember Dr Smith's address, but he lives in Apple Valley apartments at 15 Grove St and his mailbox is the fourth from the right in front of the complex." This specifies Dr Smith's address relative to a base address, 15 Grove St. In a computer, such a base address might typically be stored in an index (or general purpose) register as shown in figure 4. The displacement or address modifier in this case would be 3, which added to 15 gives the actual address of 18 Grove St. A computer with this single register indexed addressing method carries out the same form of calculation to produce the effective address: It adds the displacement or modifier field to the contents of the index register identified in the instruction.

An absolute or direct address specifies an operand location as a fixed number embedded in the instruction sequence.

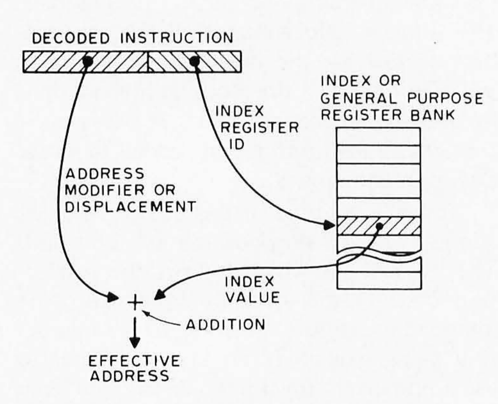


Figure 4: Indexed Addressing. One common mode of addressing is called indexed addressing, in which an index register specifies one numeric value which is added to an address modifier to produce the effective address. If the index register contains a base address value, then the modifier specifies a displacement or offset which is added to the base; if the index register contains an offset or displacement, then the modifier field is interpreted as a base address. In either case the result is an effective address.

Use of registers for address components enables one to employ base and index address concepts.

In the most general case, the index register may contain either an actual base address such as the first address of a table of values, or a displacement value. The corresponding contents of the modifier would be a displacement value or a base address, respectively. In some presently existing microprocessor designs, the index register is not large enough to contain a *full* base address. For instance, this occurs if the microprocessor uses a 16 bit address space and contains only an 8 bit index register. This case would require using the index register to contain a displacement with the base address becoming the instruction's modifier field.

Other options which sometimes occur include the choice of a second register as a component of effective address generation. In such cases, the instruction specifies one register which is intended as a base register, and a second register which is intended as an index register, as shown in figure 5. This form of double *register addressing* is sometimes combined with a modifier field as shown in figure 5. At this time, however, the microcomputers commonly available do not have such a powerful addressing mode.

One of the advantages of using a base register as well as an index register is that the base register can be used to locate a segment of memory, while the index register is used to access various places in that segment according to the program. Since all addressing is specified relative to the base register, relocating the program or data being referenced can be accomplished without modifying any code except the instructions which load the base register. The example of figure 6 shows the case of a computer which specifies a jump instruction effective address as the sum of a base register (register 0) and a displacement. Loading the same binary code at location 100 or 1125 is possible, provided the base register is initialized at the start of the program. The problem of relocation thus consists of redefining the constant which will be loaded into register 0 at the start of the program.

Program Counter Relative Addressing

Program counter relative addressing is very similar to indexed addressing except that the base address is implicitly specified using the program counter. In a typical machine which allows program counter relative addressing for data as well as program control purposes, the instruction contains a modifier relative to the current contents of the program counter as shown in figure 7. In some microcomputers, such as the 6800, program counter relative addressing is only allowed for branch instructions, and is specified relative to the next address following the end of the current instruction.

In terms of the postal analogy, this corresponds to the mailman coming upon a letter with no street address as he is working along his route. He therefore calls the postmaster and explains his dilemma. Since

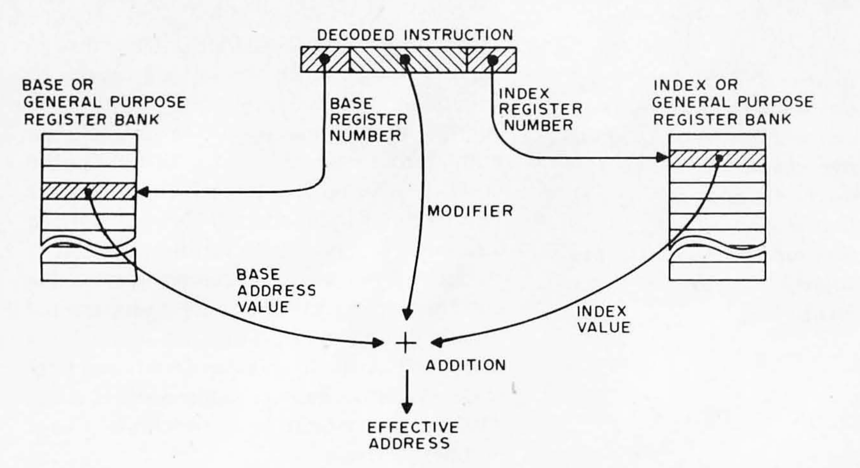


Figure 5: Combining Two Index Registers. A more general address calculation uses one register as a base register, a second register as an index register, and a modifier. The effective address is then the sum of the values found in the two registers and the value of the modifier. The order of calculation and detailed significance of the registers depends upon the processor design which uses this type of address calculation.

a.

BASE REGISTER 0: 100

100	SET REG 0: = 100
101	
102	
103	JMP 0,2
104	
105	
106	
107	

JUMP TO ADDRESS 2 PLUS REGISTER 0

b. SAME PROGRAM, RELOCATED

BASE REGISTER 0: 1125

1125	SET R0: = 1125
1126	
1127	
1130	JMP 0,2
1131	
1132	
1133	
1134	

JUMP TO ADDRESS 2 PLUS REGISTER 0

Figure 6: A base register scheme allows convenient relocation of code. In this example, the target address of a JMP (jump) instruction is specified as a base address register and a displacement. The value of the displacement is shown as two words from the start of a block of memory in which the program resides. With the base register loaded to the starting address, it does not matter where the block is located. At (a) it is located at octal address 100; at (b) the block is located at address 1125. With base addressing schemes, the first operation on entry to a program or block of code is to establish the value in the base register, as illustrated in these examples.

there is only one phone booth on the route, the postmaster gives him directions, such as: "Walk down the street directly in front of you and deliver the letter to the fourth mailbox in the apartment complex." Note that the base address is implicitly specified since the postmaster knows the location of the phone booth.

Indirect Addressing

To illustrate still another method of addressing, assume that Dr Smith recently had a post office box, #35. Since then he changed his mind and asked to have all his mail forwarded to his Grove St address. In order to remember the change when mail comes to the old address, the postmaster might mark Dr Smith's Grove St address on box 35. Then, when the mailman attempts to insert a letter for box 35 into that box, he sees the note that tells him to forward the letter to 18 Grove St. Thus, the box is not the final destination of the letter; in fact, it contains only an address to which the letter is to be forwarded. We call this method of locating the effective address (18 Grove St) indirect addressing. Figure 8 illustrates how the effective address is used to retrieve a second effective address in the computer form of indirect addressing. In the simplest form of indirect addressing, only one such level of indirection is involved.

We could easily extend this notion to multiple levels. In the postal analogy, imagine that Dr Smith moves out of 18 Grove St. The change of address order to the post office would result in a note to the postman on the 18 Grove St route, giving the new address of Dr Smith. Then, if a

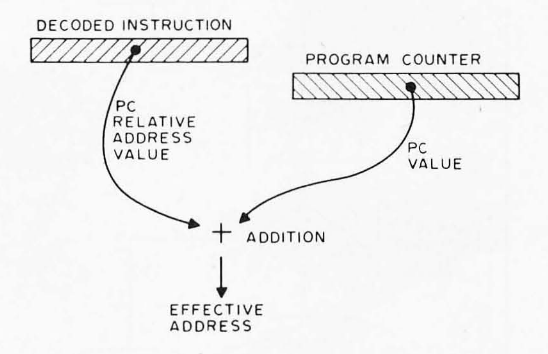


Figure 7: Program Counter Relative Addressing. Some computers provide a means to address memory in terms of an address displacement relative to the current program counter value. The instruction contains the displacement which the processor adds in the current program counter value for this type of effective address calculation.

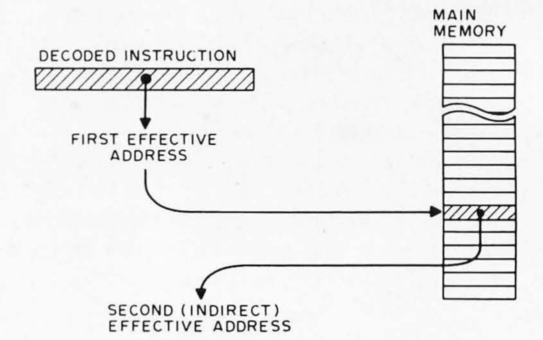


Figure 8: Indirect Addressing. In this form of addressing, the first effective address developed is used to address memory to find a pointer which will become the final effective address used for the instruction.

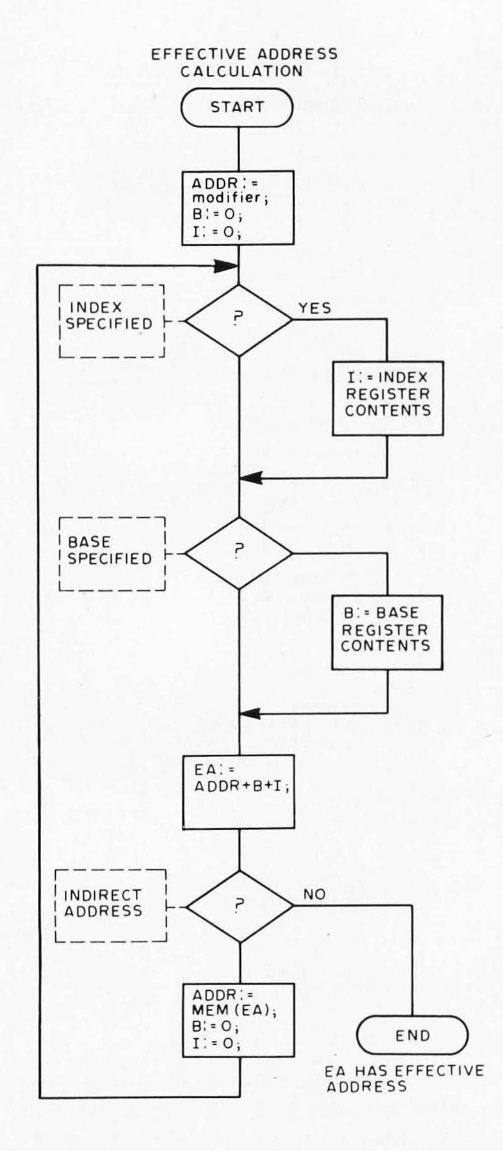


Figure 9: A General Address Computation Algorithm. This flow chart shows a typical address calculation algorithm of a modern general purpose computer. The typical microcomputer design circa early 1976 does not employ such a powerful addressing algorithm, but future improvements in chip designs should yield addressing techniques which approach the power of a good general purpose computer's addressing.

letter came to the original post office box 35 address, the postman would look up the 18 Grove St address. At the 18 Grove St address, the postman would in turn find the pointer to a new address for Dr Smith. The letter in this case would reach Dr Smith after two levels of indirection. This might happen a number of times if Dr Smith has a habit of frequently moving. In a microprocessor, the current chip designs offer only a very limited version of this mode, if indirect addressing is permitted at all. In minicomputers and large scale systems, indirect addressing is often allowed to continue to an indefinitely large number of levels.

General Address Evaluation Algorithm

Indirect addressing is often combined with the other addressing modes in computers which feature the most powerful effective address calculations. For instance, the indexed addressing mode might be used to develop the effective address for the first indirect address in a chain of indirect addresses. Once the chained indirect address lookup is begun, the processor might continue through multiple levels of indirection until a chain termination condition is detected. A general address evaluation algorithm which combines base register, index register and the possibility of indirection is shown in figure 9. Such an algorithm is typical of a good minicomputer, but is only partially implemented for most presently available microcomputer chip designs.

Summary

These methods of addressing are usually referred to as the addressing modes of the computer. To recap, the typically available modes are:

1. Immediate Addressing, in which the data being referenced forms a part of the actual instruction.
2. Direct or Absolute Addressing, in which the address of the operand is actually given as part of the instruction.
3. Indexed Addressing, in which one or more registers are specified, possibly including a modifier field. The effective address is a sum of the contents of the addressing registers and the modifier.
4. PC Relative Addressing, in which the program counter acts as a base address with an offset specified by the instruction.
5. Indirect Addressing, in which one of the other modes develops an effective address at which a pointer to data will be found.■

What's New?

Lowell Institute School – MIT

Boston area residents should check out the Lowell Institute School, affiliated with Massachusetts Institute of Technology, an evening school with professional instruction in areas of modern technology, which now includes microprocessors.

The Lowell Institute School was established at MIT in 1903 to provide instruction in technical subjects for residents of the Boston area. Today the School continues this tradition by offering subjects in the areas of modern technology which are not readily available at other evening institutions. Instruction level is geared to the practicing technician who has an associate degree or equivalent experience.

For information, contact:

Lowell Institute
MIT Room 5-118
Cambridge MA 02139
phone (617) 253-4895

A new course in the Spring 1976 catalog is "Introduction to Microprocessors," with hands on lecture and laboratory sessions dealing with available microprocessors, programming of ROMs, assemblers, debugging aids, interfacing techniques, AD conversion. Students design an interface and generate software for a device of their own choosing.

Of Course, of Course. . .

A firm called Creative Computer, 1901 Old Middlefield Way, Suite 4, Mountain View CA 94043, has come out with a "complete microcomputer software course designed to meet the needs of hardware designers who are learning software." According to the press release, the course contains 10 self teaching lessons with additional sections on computer architecture, operation and software systems. "Throughout the course, emphasis is placed on understanding the hardware/software relationships that must be considered when designing a microprocessor based digital system." The course is $49.95 FOB Mountain View, distributed in a five part sequence. Delivery is quoted 30 days ARO for the first module of the sequence. For more information phone (415) 961-5240.■

Product Description:

SCELBAL

Here is a new product description of the SCELBAL language, supplied by its authors, Nat Wadsworth and Mark Arnold of Scelbi Computer Consulting, Inc, 1322 Rear–Boston Post Rd, Milford CT 06460. Scelbi's philosophy of software provides an excellent model of a fully supported product: The firm supplies a completely documented package including user level documentation, detailed listing and program design information. By providing such complete detail, this type of product becomes an excellent package for both the novice and advanced programmer. The novice can begin with the user level documentation, using the package; the advanced programmer can use the detail design information to customize and enhance the package to suit personal tastes. . . . **CH**

Nat Wadsworth
Mark Arnold

The goal of about 90 percent of small systems owners appears to be the achievement of a system with some form of IO and enough memory to support a high level language. In assembling a system with a high level language capability there are a number of alternative paths. One common approach is to purchase a complete system from a single source with the high level language capability either bundled or as an extra cost option. However, when sole source purchases are made, there is no room for shopping around to find the best buy in peripherals, memory, software, etc. The alternative of integrating a system from subsystem components is attractive when hardware prices are totalled up, but where can one go to acquire the software needed for the custom system? One answer is to treat the software as another type of subsystem component and seek an independent supplier of high level language capability appropriate for the custom system.

When seeking a software package, one of the most important criteria is full documentation of the design at both the user level and the systems programming level. A large program without a detailed program logic description and listing can be likened to a complicated electronic device without a schematic or logic diagram.This is especially so for complicated systems programming packages such as interpreters, compilers, assemblers, monitors, etc. Installing such a program in a custom system virtually requires modifications of some form to fit the hardware details. But attempting to modify or enhance a complex program is a risky, frustrating and often downright impossible task without the detailed documentation.

SCELBAL – A Higher Level Language for 8008/8080 Systems

Few "canned" programs can be tailored to have all the features desired by all the possible potential users. To attempt to do so would result in programs requiring more memory than users could afford. The answer to this problem is, of course, to supply programs in such a manner that they can be readily modified and altered by their owners. This means, simply, that the detailed source listing for the program must be made available to the purchaser. Assisting the program owner by also providing detailed comments with the listing, a general overview of the program's organization and operation, and general flow charts can further enhance the value of the program to the owner. With this information available, the program's owner can safely proceed to tailor the capabilities of the program to serve his or her particular interests and requirements.

This is the approach Scelbi Computer Consulting, Inc, has taken in presenting its new higher level language for 8008/8080 machines. The language has been given the name SCELBAL for SCientific ELementary BAsic Language. As the reader can easily surmise from the title, it is similar in capabilities to the highly popular BASIC language. SCELBAL was specifically

developed to be able to run on 8008 based microcomputers. It is believed to be the first such higher level language to be made generally available that is capable of running in a system equipped with the ubiquitous 8008 processor. The program can of course also be run on systems using the more powerful 8080 processor though it is not as memory efficient as it could have been if the program had forsaken 8008 capability.

The language was developed to operate in an interpretive mode [See Ted Nelson's "The Magic of Computer Languages", April BYTE, page 24]. This means that the entire language processing program resides in memory at one time along with the program written in the higher level language that is to be executed. When the interpreter is given the RUN command, it immediately proceeds to interpret each line of the higher level language program and perform the necessary calculations and functions. This differs from a compiler which would first convert the higher level language source listing to machine code, then later execute the machine code.

A compiler oriented high level language generally is cumbersome to run on a small system that lacks reliable high speed bulk memory storage facilities. For instance, if the program had been designed as a compiler, the following steps would have been necessary in order to execute a higher level language program:

- First one would have to load a text editor program into the computer and create the desired higher level language version of a program as a source listing.
- A copy of the source listing would then have to be saved on an external memory medium.
- Next, a portion of the high level language system, the actual compiler, would have to be loaded into memory. When it was resident, one would produce the desired machine code version of the higher level language statements by having the compiler process the source listing several times (much as an assembler program would process the mnemonic listing when programming in machine language). The machine code produced would have to be stored on an external memory device at this stage.
- Finally, the run time portion of the high level language system would have to be loaded into the computer along with the machine code produced by the compiler.
- The higher level language program would then finally be ready to run.

An error in the original source coding for the program cannot be detected until run time. In that event you would have to go all the way back to the text editor program to correct the higher level language source listing and start the process over again. While the compilation process causes no great trouble with huge systems and plenty of high speed mass storage, it can be inconvenient compared to interpreters in small systems.

What's in the SCELBAL Program

Developing a high level language as an interpreter eliminates the requirement for the constant use of an external bulk memory device in order to get a program from the concept to execution stage. An interpreter is definitely a much more practical high level language concept for the small systems user. The entire interpreter program resides in memory at one time. An area is set aside in memory to hold the higher level program. An executive portion of the program allows the user to enter the higher level language listing directly into the area where it will be operated on when the program is executed. The executive of the SCELBAL interpreter for example provides for the user entering a program from a manual input device such as a keyboard. Or, if the user desires to run a program that has been developed previously, a LOAD command will direct the program to read in a program from an external bulk memory device such as a magnetic tape peripheral.

SCELBAL has been designed so that it can operate in a calculator mode or operate in a stored program mode. In the calculator mode, each statement is executed immediately after it is entered by the input device. In this mode, the program is ideal for solving simple formulas when the user only needs to obtain a few values.

When operating in the stored program mode, the interpreter will follow an entire series of instructions as directed by the higher level program. To enter a program that will be operated on as a stored program, the operator simply assigns a line number at the beginning of each statement.

Executive Commands

The executive portion of the package allows the user to edit a program at any time. Lines may be deleted and new lines entered anywhere in the program. If the operator makes a clerical error while enter-

ing a line, a special erase code may be used to effectively backspace within a line and then re-enter the correct characters. Furthermore, the executive checks for various types of syntax errors as statements are entered, and will display a two character error code to the programmer when such errors are detected.

The executive portion of SCELBAL has five major commands available to the operator:

- SCR (for SCRatch) effectively clears out any previous program stored in the program buffer along with any variable values.
- LIST causes the present contents of the program buffer to be displayed for review or to make a copy for record keeping if a printing device is in use.
- RUN causes the higher level language program stored in the program buffer to be executed by the interpreter.
- SAVE. This command directs the program to save a copy of the program stored in the program buffer on the user's external bulk storage device. A program saved in this manner can later be restored for execution by using the following command.
- LOAD. This command causes the program to read in a copy of a program from an external device that was previously written using the above SAVE command.

SCELBAL Statements

A higher level language program is made up of statements that direct the machine to perform selected types of operations. The SCELBAL language can execute 12 different types of statements. In addition, the END statement is used to signify the end of a program:

- The REM (for REMarks) statement indicates a comments line which is ignored as far as program execution is concerned. Information on a remarks line is intended only for the use of programmers and is used to document a program.
- The LET statement is used to set a variable equal to a numerical value, another variable, or an expression. For instance the statement:

  ```
  LET X = (Y*Y + 2*Y – 5)*(Z + 3)
  ```

  would mean that the variable X was to be given the value of the expression on the right hand side of the equal sign.
- The IF combined with the THEN statement allows the programmer to have the program make decisions. SCELBAL will allow more than one condition to be expressed in the statement. Thus:

  ```
  IF X <= Y THEN LL
  ```

  states that if X is less than or equal to Y then the program is to go directly to line number LL. Otherwise, the program is to continue on to the next statement in the program.
- GOTO directs the program to jump immediately to a specified line number. The GOTO statement is used to skip over a block of instructions in a multiple segment or subroutined program.
- The FOR, NEXT and STEP statements allow the programmer to form program loops. For example, the series of statements:

  ```
  FOR X = 1 TO 10
  LET Z = X*X + 2*X + 5
  NEXT X
  ```

  would result in Z being calculated for all the integer values of X from 1 to 10. While SCELBAL does not require the insertion of a STEP statement in a FOR – NEXT loop, a STEP value may be defined. The implied STEP value is always 1. However, it may be altered to be an integer value other than 1 by following the FOR range statement by the STEP statement and a parenthesis containing the STEP size. Thus:

  ```
  FOR X = 1 TO 10 STEP (2)
  ```

  would result in X assuming values of 1, 3, 5, 7 and 9 as the FOR – NEXT loop was traversed.
- GOSUB is used to direct the program to execute a statement or group of statements as a subroutine. The statement is used by designating the line number in the program where subroutine execution is to begin.
- The RETURN statement is used to indicate the end of a subroutine. When a RETURN statement is encountered, the program will return to the next statement immediately following the GOSUB statement which directed the program to the subroutine. SCELBAL permits multiple nesting of subroutines in a program.
- DIM (for DIMension) is used to specify the formation of a one dimensional array in a program. Up to four such arrays having a total of up to 64 entries are permitted in a program when running SCELBAL. The statement:

  ```
  DIM K(20)
  ```

  sets up space for an array containing 20 entries. (Array size must be

designated by a numerical value, not a variable.) The DIM is an optional statement that may be left out of the program to provide additional program storage space in systems having limited memory.

- INPUT is used to cause the program to wait for an operator to INPUT information to the program. After the information has been received, operation of the program automatically continues.
- PRINT is used to output information from the program. Using the PRINT statement the user may direct the program to display the value of variables, expressions, or any information such as messages. The PRINT statement allows for multiple mixed output on a single line, and the option of providing a carriage return and line feed after outputting information or suppressing that function. For instance, the statement:

  PRINT 'X IS EQUAL TO: ';X

  would result in the program first printing the message "X IS EQUAL TO:" and then the value of the variable X on the same line. After the value of the variable had been displayed, a carriage return and line feed combination would be issued. To suppress the printing of the carriage return and line feed the programmer would merely include another semicolon at the end of the statement. A comma in a PRINT statement will direct the output to start at the next tab point in a line. A special function may also be called upon to direct the output to begin at a specified position in a line to allow for neat formatting.

The power of the language is further enhanced by the inclusion of seven functions that may be used in statements. The seven functions available in SCELBAL are:

- INT returns the integer value of the expression, variable, or number requested as the argument. This is the greatest integer number less than or equal to the argument.
- SGN returns the sign of the variable, number, or expression. If the value is greater than zero, the value +1.0 is returned. If the value is less than zero, the value -1.0 is returned. The value 0 is returned when the expression or variable is zero.
- ABS returns the absolute value (unsigned magnitude) of the variable or expression identified as the argument of the function.
- SQR returns the square root of the expression, variable, or number.
- RND produces a semi pseudo random number in the range of 0 to 0.99. This function is particularly useful to have available for games programs.
- CHR is the character function. It may be used in a PRINT statement and will cause the ASCII character corresponding to the decimal value of the argument to be displayed. (A reverse function is available for the INPUT statement which will return the decimal value of a character when it is inputted.)
- TAB may also be used in a PRINT statement to direct the display device to space over to the column number specified in the argument. This function allows the programmer to format the output into neat columns.

SCELBAL Background Information

User defined variables are limited to one or two characters. A variable must begin with a letter of the alphabet. Limiting variables to a maximum of two characters helps conserve memory space. Up to 20 different variables may be defined in a single program.

SCELBAL allows the use of fixed and floating point notation. A minimum of 23 binary bits are used in the mantissa portion of all calculations allowing for calculations with six to seven significant decimal digits of precision. The exponent range is from plus to minus the 38th power. Numbers may be inputted in either fixed or floating point notation. Output from the program is automatically selected to be either fixed or floating point, depending on the size of the number that is to be displayed.

The package, without the optional DIM statement, is designed to run in 8 K bytes of memory in an 8008 or 8080 system with approximately 1250 bytes for program storage. With this amount of storage available, surprisingly complex programs can be executed. The program authors have successfully loaded and run such games as Lunar Landing in this configuration by reducing the number of messages issued to the player.

The DIM statement requires approximately 750 bytes of memory. It is recommended that users desiring to include the DIM capability have more than the mini-

mum 8 K of memory available in their system. A particularly attractive feature of SCELBAL is that users with more than 8 K of memory can use the additional space for program storage. Thus, for example, a 12 K system will enable a user to execute SCELBAL programs having as many as 150 to 200 statements.

A major concern of the developers of SCELBAL was that the 8008 processor might make the language so slow that it was impractical for the user. Our tests indicate that the time to perform typical calculations, while they are slow compared with more powerful machines, are certainly tolerable. For instance, the typical response time from input to the display of a new set of parameters when running the Lunar Landing game is on the order of six to seven seconds. A program that calculates the mortgage payments on a house on a monthly basis, and displays such values as the payment number and balance after each payment, requires a few seconds between the displaying of each new line of information. A dice playing game responds with new throws of the dice in the order of a second or so when using a formula that includes the use of the random number generator. These times are by no means fast, but they are certainly adequate for the intended uses of this language on an 8008 system. The developers were pleasantly surprised with the overall speed performance of the package. Of course, these response times can be cut almost in half by using an 8008-1 processor. Naturally, if the program is installed in an 8080 system, the response time is improved an order of magnitude.

The Listing

Since the program for SCELBAL is supplied in the form of a publication that includes a complete highly commented source listing (as well as assembled object code for both the 8008 and 8080), the user who desires to modify or expand the capabilities of the basic package will be in a position to do so. It is felt that the availability of such a powerful program in this form will greatly enhance the general usefulness of small systems and open new vistas to users. The program in this form should also be of considerable value to educators who desire a good reference framework from which to introduce students to the development of similar packages.

The publication is being made available for the first time in June 1976 by the developer, Scelbi Computer Consulting, Inc, 1322 Rear – Boston Post Rd, Milford CT 06460.■

Classified Ads Available for Individuals and Clubs

Readers who have equipment, software or other items to buy, sell or swap should send in a clearly typed or printed notice to that effect. The notices are free of charge and will be printed one time only on a space available basis. Insertions should be limited to no more than 100 words. Notices can be accepted from individuals or bona fide computer users' clubs only. Commercial advertisers should contact Virginia Peschke at BYTE for the latest rate card and terms.

WANTED: Back issues of BYTE magazine to include December 1975 issue. Please send information regarding condition and asking price. Bob Thomas, 3173 Stephens Av, Ogden UT 84401.

WANTED TO BUY: JUC Video gard 5" video monitor, also EROM boards for IMSAI 8080. FOR SALE or trade: one IMSAI 22 slot expansion board with all sockets, fully functional, $150 or trade for above items. Also two MITS dynamic RAM 4 K by 8 memory boards, all ICs in sockets. Completely tested and functional, $200 each or trade for above items. Terry L Todd, Box 319, Platteville WI 53818, (608) 348-4701.

SAVE MY MARRIAGE! Buy my new assembled IMSAI 8080, loaded 22 slot mother board, 8 K RAM, regular price, $1835. Will sell to highest bidder above $1700. Also IMSAI 8080 kit, still in box, large mother board, regular price $578. Will sell to the highest bidder above $547. Send bids to 664 Via Alamo, San Lorenzo CA 94580.

FOR SALE: 33 ASR Teletype, works perfectly. $650. Contact A J Rudis, (312) 349-9002 after 6 PM.

TRADE: I have several 10 inch reels of 1/2 inch computer/video tape to trade for the first four issues of BYTE. Ed Lindberg, Bowmansville NY 14026.

FOR SALE: Altair 8800, 256 words memory, manual. Excellent condition, in full working order. $500 postpaid. Call: (312) 562-7100, Ext 752, between 8:30 & 5:00, or write: Neal Puncochar 4N725 Klein Rd, W Chicago IL 60185.

FOR SALE: (One only) Pitney Bowes core memory plane with selection diodes. 4 K by 16 bits, 3 wire. I will include prints. $45 Bruce Filgate, Box 284, Maynard MA 01754. Phone (617) 481-7400, x6618.

FOR SALE: Altair 8800 w/2 K memory, fan, P T Co 3P+S I/O, moboard, card cage, 16 V transformer, bus terminator. Good working condx. $850 value, asking $500 for package. John Moorhead, 928 J St, Davis CA 95616. Phone MTFSSUN eves (707) 758-2495.

FOR SALE: Burroughs C-3660 Programmable Calculator. With memory and magnetic card capabilities. Very good shape. First check over $700. Tom French, 909 Society Av, Albany GA 31701.

LOOKING for a used or not working minicomputer. Altair, IMSAI, C-MOD or home brew, parts, boards, keyboard, tape drive, or? Price open. Steve Kelley, 9506 Peach St, Oakland CA 94603.

FOR SALE: Redcor RC-70 midi computer kit, partially assembled, with 64 K bytes of core memory, DMA & interrupt controller cards. Capable of FORTRAN and BASIC (available). Price $1000. Also have IBM 727 Tape Drives $150 each FOB, Buster Killion, 2773 Winrock Av, Altadena CA 91001, (213) 798-2977.

FOR SALE: DATAPOINT 3300 ASCII magnetic reed keyboard with data. TTL logic, needs +5 V and clock for serial data output, with new case, base and two screens, $40. UNIVAC 1007 used keyboard with base, spare key switches and new ASCII ROM, $16. BRPE-9 5 level Teletype Corp punch, perfect and working, $20. Junker BRPE-9 missing chad box, tape holder and punch head, $7. Western Union card reader, new in original box and packing, no data. Send SASE for more info, $110 (easily worth $350). All prices negotiable, no shipping included. Neil Ferguson, 116 Woodland Cir, Arlington TX 76013, (817) 461-2867.



BOOK REVIEWS

Computer Chess *by Monroe Newborn, Academic Press, Inc, New York, 1975. $15. (Published under the auspices of the Association for Computing Machinery Inc.)*

"My microcomputer swiftly checkmated the incredibly big machine," Tom said rookingly.

Impossible? A look at the second appendix of *Computer Chess* shows that several of the older US computers and two newer foreign ones used to run chess programs had less memory than would a fully developed personal micro. In the first US championship (1970), a mini using a 4 K program of 18 bit words (roughly equivalent to 9 K of 8 bit words) came in second. Also, the typical chess program can be split into three parts for opening, middle and end games, with only one at a time in the working memory.

However, combine all the memory you can install with this book, and you'll still be a long way from the first move. The book has several useful flow charts, but no part of any program ready to load in any language. Nor, it appears, is any complete program readily available, but a more thorough checking of the many references given might turn up one with a full listing. Of the chess programs mentioned, it seems that none exists in BASIC; but two are in FORTRAN IV, one is in ALGOL and one is in PL/1, version IV. More are written in assembler languages as these are more flexible. But assembler language is not as easily translated from one machine to another.

Although this book is not written with the computer hobbyist in mind, it is more than a pleasant diversion for dilettante dreamers. But it will no more teach you how to program than it will teach you how to play chess well. In 200 pages, how could it? It is less likely to encourage your half-vast plans than to sober you with the difficulty, even enormity, of the task. It has taken skilled programmers several months, full time, at the least, to write a chess program, and about a year more to test, analyze and debug it. And if you expect it to play really well, at master level, the author predicts the need for 15 years (eight more than anyone has done to date) of intensive collaboration between an excellent programmer and a very good player. For all its speed and accuracy, the computer is not likely to play better chess than its programmer. Today's top programs have come a long way, and can give the ordinary player a good game. But so far, no program is able to improve itself by learning from its failures.

Those failures are in themselves amusing, and they are difficult to debug. The book describes one chess program which, knowing for seven moves that it could win in one or two moves, simply did not bother to do so, letting the opposing program squirm free and win.

Thirty-eight annotated games, said to be all the important ones through the first world computer chess championship in 1974, occupy the bulk of the book. They range from mercilessly short to painfully long. For many games, the time per move and number of possible moves considered at each turn is given. A historical chapter explains the essential aspects of game theory: the mini-max algorithm and the alpha-beta algorithm, and Shannon's type A and type B strategies. Later, the program OSTRICH, developed by the author and George Arnold, is described in considerable detail. An introductory chapter also states the author's conclusions. He sees a new era in chess emerging, in which the game or challenge will be in programming computers to play.

This reviewer drew other conclusions. Although some chess programming has been intended to simulate human thinking or help understand it better, the way most programs play is by considering vast numbers of possible moves, something we do not do, unless quite subconsciously. As the author says, there have been no new fundamental breakthroughs in chess programming. If such a breakthrough could reduce these vast numbers, it would certainly help put the micros across the board. Meanwhile, if you are not a hotshot programmer, don't rush to buy more memory so your micro can start

woodchopping. Start with a simpler game until you can see what an ultimate challenge chess seems to be.

Events move fast in the computer field and even chess does not entirely deserve its reputation as a slow game. I recently received Volume 4, Number 4, of *PCC*, with a letter from J G Day, 17042 Gunther St, Granada Hills CA 91344, on page 27 that the September *Consumer Bulletin* (British Computer Society, 29 Portland Place, London W1N 4HU) has a letter from M A Bramer, "who mentions that he has a sophisticated chess program running in BASIC. His program embodies a novel pattern-matching method requiring no tree-searching whatever." This could be a fundamental breakthrough.

Then, Andrew Soltis, in his column in the March 13 *New York Post,* writes of "a chessplaying microcomputer that a Brooklyn firm, Cardinal Industries, expects to sell within the year for about $120. Considering what happened to pocket calculators, we might expect a time within the near future when anyone can afford an electronic opponent."

The hobbyist starting now may not only be outprogrammed by an over-the-counter machine before he gets running well, but may have his cost undercut by more than ten times. How's that for a challenge?■

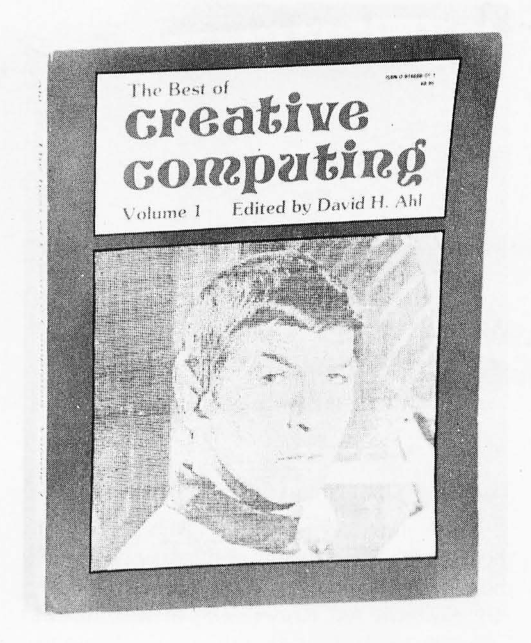
John F Sprague
Allendale NJ

The Best of Creating Computing – Volume 1, *David H Ahl (editor), Creative Computing Press, Morristown NJ 07960; 1976; 8-1/2 x 11, 328 pages. Paper $8.95.*

The Best of Creative Computing – Volume 1 contains articles and fiction about computers, games that you can play with computers and calculators, hilarious cartoons, vivid graphics and comprehensive book reviews.

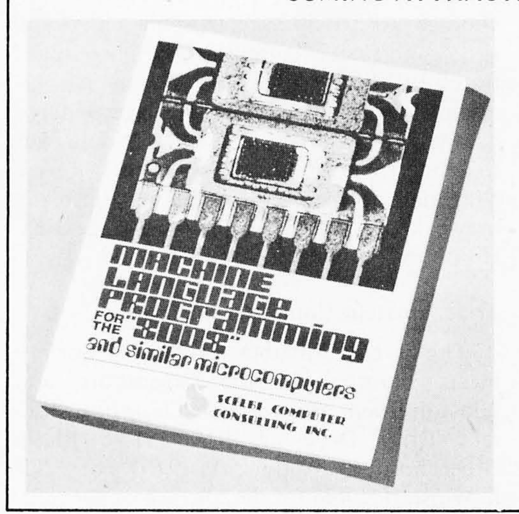
Authors range from Isaac Asimov to Senator John Tunney of California; from Marian Goldeen, an eighth grader in Palo Alto, to Erik McWilliams of the National Science Foundation; and from Dr Sema Marks of CUNY to Peter Payack, a small press poet. In all, over 170 authors are represented in over 200 individual articles, learning activities, games, reviews and stories.

This 328 page book has 108 pages of articles on computers in education, CAI, programming, and the computer impact on society; 10 pages of fiction and poetry including a fascinating story by Isaac Asimov about all the computers on earth linking up



after a nuclear war to support the few remaining survivors; 15 pages of "Foolishness" including a cartoon piece called "Why We're Losing Our War Against Computers"; 26 pages on "People, Places, and Things" including "The Compleat Computer Catalogue" which gives capsule reviews and lists sources for all kinds of computer related goodies; 79 pages of learning activities, problems and puzzles; 29 pages containing 18 computer games including a fantastic extended version of one of the most popular computer games: Super Star Trek; and 32 pages of book and game reviews including Steve Gray's review of 34 books on the BASIC language. *The Best of Creative Computing – Volume 1* is currently available by mail only for $8.95 plus 75¢ postage from Creative Computing Press, PO Box 789-M, Morristown NJ 07960.■

COMING ATTRACTION



Scelbi Computer Consulting, Inc, 1322 Rear Boston Post Rd, Milford CT 06460, has prepared an excellent book entitled Machine Language Programming for the "8008" and Similar Microcomputers. *We won't go into complete detail at this point; however, BYTE readers will be treated to a series of three excerpts starting with Chapter 1 in BYTE's July issue.*■

Clubs and Newsletters



The New York Amateur Computer Club

This club, organized by Bob Schwartz, meets on the second Thursday of each month at Manhattan Community College, 799 Seventh Av, Room 605, 6 to 9:30 PM. From 6 to 7:30 is the time for demos by companies and club members, and a trading period. At 8 the meeting has speakers on topics of interest to the members. In March the club considered the new organizational by laws and set up committees. In April, "we will approve the by laws and get down to business."

Allen Yoricks is conducting a class for members who want to study and obtain their amateur radio licenses. The aim is to use ham radio to communicate with other computers in distant parts of the world. In a reverse twist, the members of a Brooklyn Ham Radio Club want to learn about computers for the same purpose. They want to do computer time sharing across the world via a satellite!

For further information, contact Stanley Veit of the Computer Mart, 314 Fifth Av, New York NY 10001, (212) 279-1048.

Anchorage Alaska

Ronald J Finger, 3417 E 65th Av, Anchorage AK 99502, reports an active and very informal computer/amateur radio fraternity in Anchorage. While he has his doubts about whether the rugged individualists who inhabit his city would get together to create anything so formal as a club, he's agreed to point the way into the network of contacts in the Anchorage area. His phone number is (907) 344-6503.

Triad Amateur Computer Society

The Triad Amateur Computer Society meets monthly in the Greensboro and Winston-Salem areas of North Carolina. Contact Doug Drye at (919) 373-0040 in Greensboro, or Andy Pitts (919) 765-1277 in Winston-Salem for details.

Montreal Club?

I've been waiting for an announcement concerning a Canadian computer user's society. Not having seen any up to now in your Clubs and Newsletters section, I propose that interested persons in Canada contact me so we can set one up. I would be ready to coordinate such an effort. I am an electronic engineer and am presently studying for an MBA at McGill University. I have built a CT1024 TV terminal and am presently building up a 6501 based microprocessor system.

Leslie Zoltan
4100 Kindersley St #22
Montreal, Quebec
(514) 733-8890

Rochester NY Club Activities

Peter Helmers reports on the creation of a microcomputer club for the Rochester NY area. An interest meeting April 1 at the University of Rochester drew 36 persons after arrangements were made by an ad hoc steering committee. Meetings are to be held every four weeks, with newsletter subscription dues set at $2 per year. Affiliation with the SCCS is being considered. For further information, contact:

University of Rochester
Computing Center
727 Elmwood Av
Rochester NY 14620

Att: Microcomputer Club of Rochester

Long Island Computer Association

Gerald S Harrison sent in a note about the latest activities of the LICA:

> "A word about the club . . . . Friday night [February 20 1976] was our first open general meeting at our permanent meeting location. We were thrilled at the turnout, approximately 80 people, many heavily into computers. Thirty-four of the group work with computers, 35 know computer languages and at least 20 of the group I would rate as professional software types. Fifteen members have machines, one even brought down an IMSAI 8080; it looks great. Motorola demonstrated a 6800 and gave a talk about it.
>
> For future reference, meetings will be held on the third Friday of the month at 8 PM at the New York Institute of Technology, Building 500. We will endeavor to reschedule meetings that fall on holidays."

Along with Gerald's letter came a copy of

The Stack, Volume 0 Number 0. This is the first issue of the LICA's official newsletter, edited by Morris Balamut, PO Box 864, Jamaica NY 11431. For individuals desiring the latest LICA information, call Gerald Harrison at (516) 938-6769 (evenings) or Ken Kaplan at (516) 781-9859 (7:30 PM to 3:30 AM).

Ventura County Club

Doug Penrod of Santa Barbara CA sent in a note mentioning the existence of a new club for Ventura County, California, which met in Oxnard for the first time January 24. Present were Ward Spaniol (president) and Art Childs (*Interface* editor) of the SCCS. Forty people were present at the first meeting, including ten 8080 users (mostly Altairs, one or two IMSAIs and some home brewers). Many of the people who showed up were also amateur radio operators. For information on the Ventura County Club, contact Eric Strohbehn, 4409 Vineyard, Oxnard CA 93030.

ON LINE

D H Beetle's *ON LINE — Hardware & Software Exchange* is progressing towards its goal as a "buy and sell forum for the computer hobbyist." The latest issue received at BYTE was Volume 1, Issue 03, with five pages of commercial and non-commercial classified advertisements. Subscriptions are $1 for four issues, $3.75 for 18 issues or $7 for 36 issues (higher rates for foreign surface and airmail delivery classes). Contact ON LINE, D H Beetle, Publisher, 24695 Santa Cruz Hwy, Los Gatos CA 95030.

MIKE Users Group

James W Farschon, 3949 Mt Everest Blvd, San Diego CA 92111, sent BYTE a small sampling of the MIKE 2 INFORMATION PACKET Number One which according to his form letter contains 60 pages of software listings useful to 8008 owners.

> "Our real hope is that his info packet will provide the impetus for the formation of an active national MIKE user organization. Some of my ideas on the organization and activities of such a group are contained in the Preface" of the packet.

Contributors to the first packet include Mark A Condic III, Eric Schott, Jim Farschon, Tom Kasper and Jim Tucker. The purpose of the MIKE Users Group is dissemination of MIKE information to hobbyist clubs and publications, publication of further INFORMATION PACKETS with user contributions, and group projects such as system configuration, software development, etc.

Information in the first information packet includes the following titles (partial list):

Theory of IO Interfaces (Condic)
CREED Parallel Input Interface (Condic)
MIKE 2 Hardware Push Pop Stack (Schott)
Super NIM Game (Farschon)
Keyboard Monitor Program (Tucker)

The MIKE 2 INFORMATION PACKET Number One is available for $5 from Jim Farschon.

Indianapolis Club?

Keith A Pieper, 54 Sherry Ln, Brownsburg IN 46112, would like to contact individuals interested in forming a computer club in the Indianapolis IN area.

New England Computer Society

The organization of the New England Computer Society is settling down onto a regular basis. At an executive committee meeting March 10, volunteers for editing of the society's newsletter were present and duly appointed. Editor is Bob Tripp, 8 Fourth Ln, South Chelmsford MA 01824. He can be reached by phone at (617) 275-8300 (days) or (617) 256-3649 (evenings). Assistant editors are Calvin Moerrs,

Rockford Research Inc, 140½ Mt Auburn St, Cambridge MA 02138, (617) 876-6776 and Jeff Siegel, (617) 667-3111, extension 3195 (days), or (603) 635-7404 (evenings).

The NECS mailing address is PO Box 198, Bedford MA 01730, and meetings are held on the first Wednesday of the month at the cafeteria, Building C, of the Mitre Corporation, Bedford MA (junction of Routes 3 and 62).

News of CACHE

The Volume 1 Number 2 issue of the *CACHE Newsletter* (PO Box 36, Vernon Hills IL 60061) described happenings in the Chicago area. Technical information in the newsletter included a short note about the Zilog Z80 "super 8080" chip, a set of software notes by Ward Christensen, and a "Basic Computer Hobbyist's Library" listing with short descriptions of several books. Tentatively scheduled future meeting topics listed in this issue included:

> May Meeting: Ted Nelson, author of *Computer Lib/Dream Machines,* giving a talk.
>
> June Meeting: Computer Fest – come sell/buy/swap equipment and information.

For current information contact CACHE at its mailing address or call William T Precht at 620-1671.

ACGNJ News

The March issue of the *ACGNJ News*, Volume 2 Number 3, carried a report of the February 20 meeting which included demonstration of a Southwest Technical Products 6800 processor owned by the Union County Technical Institute, and a presentation of the TV Dazzler peripheral (see "About the Cover," page 6) given by Tom Kirk and Roger Amidon, using an IMSAI-8080 processor and a color TV monitor loaned by Union College.

The 8080 Sub Group of ACGNJ is handled by Dennis Dupre, who can be reached at (201) 688-9254. It meets separately to exchange information among owners of 8080 based systems.

The address of ACGNJ is:

Sol Libes, ACGNJ
Union County Technical Institute
1776 Raritan Rd
Scotch Plains NJ 07076

Membership dues are $2.

Philadelphia Activities

Richard Moberg, Philadelphia PA, sent in two items for this department:

"1. We are starting a computer society in the Philadelphia area for amateurs, students, professionals, etc, for the purpose of information exchange and education in all aspects of computers. Interested individuals should contact me at 404 S Quince St, Philadelphia PA 19147, or call (215) 923-3299 (evenings).

"2. I am working on several applications of microcomputers in medicine and would like to hear from others doing the same or with ideas for applications. Please contact me at Dept of Neurosurgery, Jefferson Medical College, Philadelphia PA 19107, (215) 829-6744."

Peoria Activities?

James Hull, 502 Joliet Rd, Marquette Heights IL 61554, is interested in starting a club in the Peoria IL area. Interested parties should drop him a line.

News of DACS

The Volume 1 Number 5 issue of the *Denver Amateur Computer Society Newsletter* reported on recent activities in that city. Scheduled for the March 17 meeting was a talk by Dr Robert Suding entitled "Comparative Hardware and Software Analysis of 8080 versus 6800 versus 6500," with a demonstration of all three processors to illustrate concepts of program transferability.

The *DACS Newsletter* also announced the activities which will be jointly sponsored by DACS and the ARRL at the American Radio Relay League's amateur radio convention in July. The two concurrent technical sessions scheduled for Friday July 16 will have microprocessors as the primary theme:

Introduction to Microprocessors for Beginners (Grand and Junior Ballrooms, Hilton Hotel, downtown Denver).

- 2 PM: Demonstration of Microprocessors in Amateur Radio Applications
- 3 PM: What is a Microprocessor?
- 4 PM: What is so Hard about Hardware, and is Software Really Soft?

Microprocessor Topics – Advanced (Assembly Rooms 2 and 3, same hotel).

- 2 PM: Putting Your Microprocessor to Work in Your Amateur Station
- 3 PM: Comparative Analysis of Microprocessor Architecture
- 4 PM: Advanced Software

The evening session, 7 PM to 11 PM in Assembly Rooms 2 and 3 of the Hilton, will be devoted to further informal discussions of

microprocessor topics and demonstrations, conducted by the speakers at the afternoon sessions.

Early reservations are recommended for accommodations. Advanced registration for the three day program is $4 until June 30, $5 after June 30. Registration forms are available from DACS or ARRL. Contact DACS at PO Box 6338, Denver CO 80206.

ARRL Atlantic Division Convention

The Bicentennial Amateur Radio Convention of the ARRL Atlantic Division will be held July 23-25 1976 at the Ben Franklin Hotel in Philadelphia. According to Harry Brown, WA3NGK, one of the coordinators of technical sessions at the convention, there will be a stress upon digital electronics and the use of small processors for amateur radio applications. Interested parties should contact the ARRL for details about the show.

LUMP is All Together

The LUMP (Louisville Area Users of Microprocessors) computer club has been formed in Louisville KY. Anyone interested in club activities in the Lousville area is invited to attend biweekly meetings. Membership circa March 15 1976 was approximately 30. Members are working with 6800, 6502, 8008, 8080, PACE and LSI-11 designs; at least one 8080 or 8008 multiprocessor system is in the works, and a club system is being built using a 6502 chip. Contact either of the following individuals for further information:

> Steve Roberts, Cybertronics, PO Box 18065, Louisville KY 40218
> Andy Ehalt, 115 Edgemont Dr, New Albany IN 47150

Tampa FL Activities

Donald A Marsh sends word of the Microcomputer Society of Florida which has 48 members meeting in the vicinity of Tampa FL.

> "The club is interested in hardware and software of all microcomputers and everyone in Florida is invited to join. Chapters are being formed in Jacksonville, Miami and Orlando. We have an active group of progressive people with discussions on microcomputer topics presented by specialists in the field."

For further information, contact Donald at 5405-B, Southern Comfort Blvd, Tampa FL 33614.

Chicago Ham Hackers Take Note

Robert C Nutting, K9TXS, would like to get together with radio amateurs in the north and northwest sections of greater Chicago IL, persons who are also into computers. His address is 6641 Palma Ln, Morton Grove IL 60053.

Northwest Computer Club

The Northwest Computer Club has mushroomed from a meeting at the house of Bob Wallace January 12 into a full fledged club which meets at 7 PM the first and third Tuesdays of each month, usually at the Pacific Science Center.

NCC Newsletter editor is Bob Wallace. Volume 1 Number 1 contained a short account of the club's history through its March meeting schedule, a list of members' names and addresses and interests, and some technical comments about graphics standards. Also present were several "short and sweet" 8080 routines supplied by Bob Wallace.

NCC Newsletter address is PO Box 5304, Seattle WA 98105. Club address is Northwest Computer Club, Pacific Science Center Foundation, 200 2nd Av N, Seattle WA 98109.

Computer Hobbyist Group – North Texas

The March issue of the *CHG-NT Newsletter*, Volume 2 Number 3, had a lot of technical information as part of its seven sheets of reduced Xerographic copy. Summary of the February 21 meeting reported a presentation by John Lawrence on "Microcomputer Applications to Amateur Radio." John demonstrated two Model 28 Teletype machines in interactive operation. Ralph Tenny provided an excellent review of the *MOS Technology Microcomputer Handbooks*, and Bill Fuller provided some notes on "Spiraglyphics" (or the problem of figuring out what is the obscure significance of markings on surplus parts). Bill also provided some observations on wiring and construction techniques, inspired by John Lawrence's impeccable point to point soldering techniques. Also published was a "Universal Code Chart for Data Communications" supplied courtesy of Atlantic Research Corporation, a manufacturer of data communications equipment.

Mailing address for the CHG-NT is c/o Bill Fuller, 2377 Dalworth 157, Grand Prairie TX 75050.

Programming Quickies:

Do you ever spend a spare moment creating a little program or subroutine to explore some of the possibilities of your computer? Write down a symbolic and absolute listing in the language of your computer plus a short paragraph describing the program and its purpose. Then send the result to Quickies, BYTE, 70 Main St, Peterborough NH 03458. Each Programming Quickie published will earn its originator $20 worth of fame and fortune.

JITTER

Gordon M Speer
SATCOMDET Box 9
FBPO Norfolk VA 23953

Here is a little blinking lights application program which works with Altair 8800A hardware. The display is a single bit seen in the front panel address lights A8 to A15, constantly moving left or right in a random walk. The program was written by reader Gordon M Speer after he found that one of the CPU registers will show up in the address lights when delay loops are run on his Altair 8800. All numbers in this listing are octal. . . . **CH**

```
000/000 006 020       JITTER:   MVI  B,020
000/002 015           JBDELAY:  DCR  C
000/003 302 002 000             JNZ  JBDELAY    Delay loop allows register B
000/006 025                     DCR  D           to show up in Altair 8800 address
000/007 302 002 000             JNZ  JBDELAY     lines A8 to A15;
000/012 315 100 000             CALL RND        Generate random number in A;
000/015 273                     CMP  E          Is A EQ E?
000/016 137                     MOV  E,A        Move A to E always;
000/017 170                     MOV  A,B        Move B to A always;
000/020 312 033 000             JZ   RECYCLE    If A EQ E then continue;
000/023 362 032 000             JP   ROTATEA    If a GT E then rotate A left;
000/026 017                     RRC             Else rotate A right;
000/027 303 033 000             JMP  RECYCLE    And continue;
000/032 007           ROTATEA:  RLC             Rotate A left;
000/033 107           RECYCLE:  MOV  B,A        Set up for next pass;
000/034 173                     MOV  A,E
000/035 346 xxx                 ANI  xxx        Mask to alter delay length;
000/037 117                     MOV  C,A
000/040 127                     MOV  D,A
000/041 303 002 000             JMP  JBDELAY
```

RND: This is an 8080 version of the random number generator published in *The Computer Hobbyist,* Volume 1, Number 5, as an 8008 program.

```
000/100 041 146 000             LXI  H,000146   Load H,L with pseudo random code address;
000/103 016 010                 MVI  C,010      [Changed from B in original]
000/105 176                     MOV  A,M
000/106 007           RNDLOOP:  RLC
000/107 007                     RLC
000/110 007                     RLC
000/111 256                     XRA  M
000/112 027                     RAL
000/113 027                     RAL
000/114 055                     DCR  L
000/115 055                     DCR  L
000/116 055                     DCR  L
000/117 176                     MOV  A,M
000/120 027                     RAL
000/121 167                     MOV  M,A
000/122 054                     INR  L
000/123 176                     MOV  A,M
000/124 027                     RAL
000/125 167                     MOV  M,A
000/126 054                     INR  L
000/127 176                     MOV  A,M
000/130 027                     RAL
000/131 167                     MOV  M,A
000/132 054                     INR  L
000/133 176                     MOV  A,M
000/134 027                     RAL
000/135 167                     MOV  M,A
000/136 016                     DCR  C
000/137 303 106 000             JNZ  RNDLOOP
000/142 311                     RET

000/143 xxx xxx xxx   xxx             Pseudo random number "seed" (must not be zero).
```

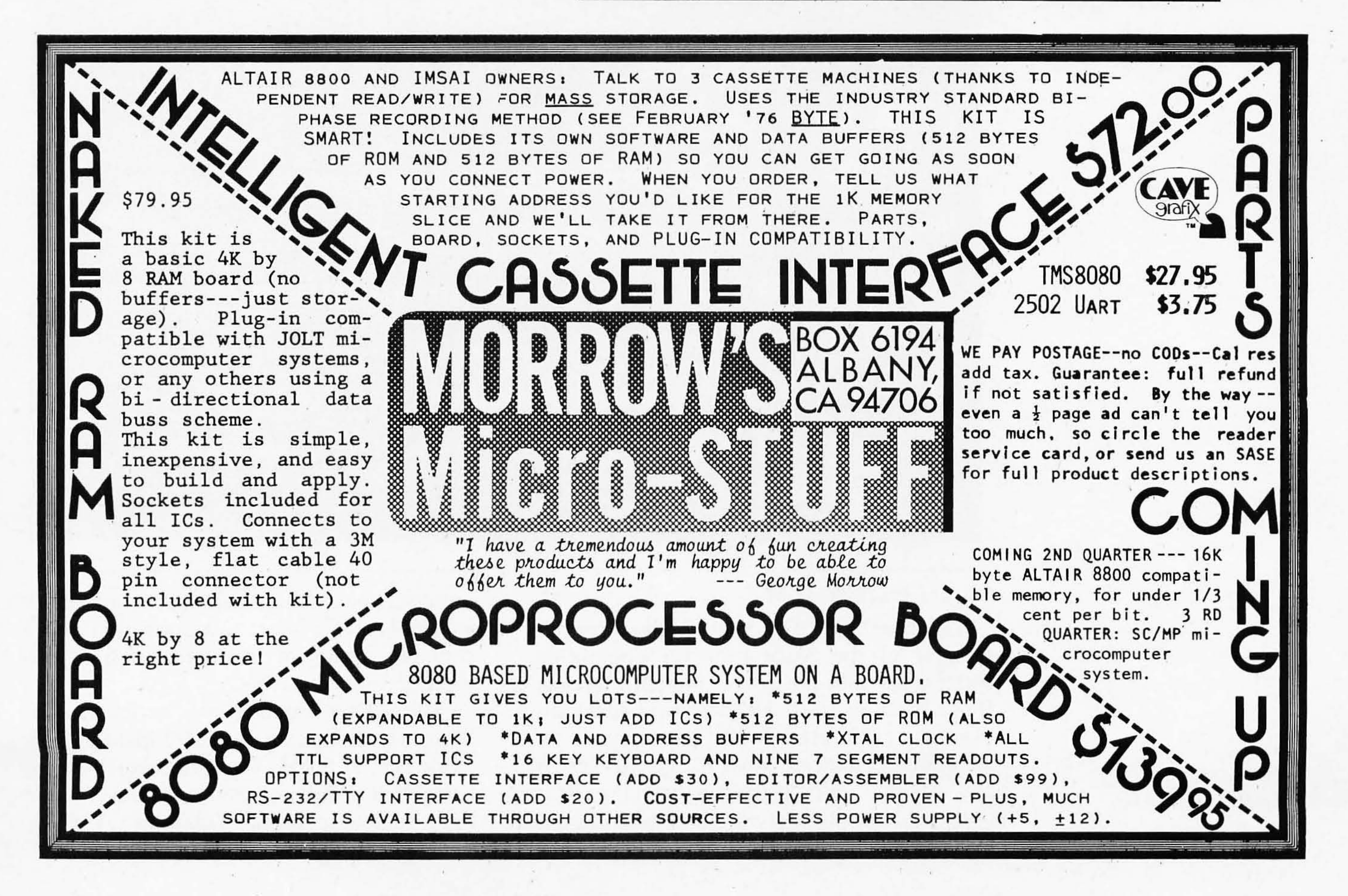
What's New?

Trekking Through Outer Space

Trek Competition™ is the name of a contest using the Trek 75™ game which is being made available via time sharing to interested computer hobbyists who have standard telephone modem equipment. Unfortunately, the press release arrived after BYTE's deadline, so the April 12-16 date of the first competition which is sponsored by GRW Systems Inc of Mountain View CA is of historical interest only.

Trek 75™ is a program written by William K Char which presents an advanced battle simulation game based on the TV series "Star Trek." The Trek Competition had an entry fee of $5 with the bulk of this money (75%) earmarked for prizes. 40% of the total entry fees becomes first prize, 25% becomes second prize, and 10% becomes the third prize. The press release indicated that the competition would be repeated at a later date. For information send an SASE (self addressed, stamped envelope) to:

Trek Competition™
2580 Westford Way
Mountain View CA 94040■



What's New?

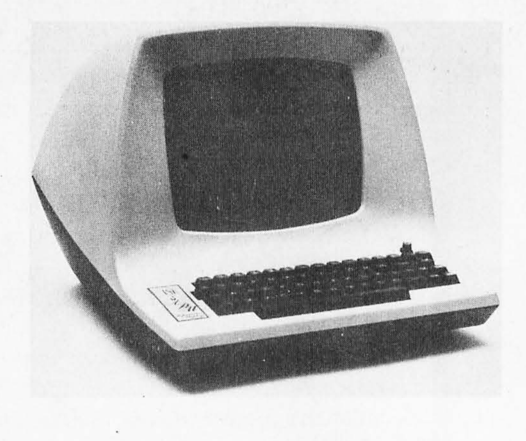


Photo 1: This is what the ADM-3 "Dumb Terminal" from Lear Seigler looks like.

It's $mart to Play Dumb

The word "dumb" has come into vogue lately to describe terminals which incorporate input and output display capability, but no microprocessor intelligence (or voiced IO). By this definition, a Teletype hard copy terminal, or a noiseless soft copy "glass Teletype" video terminal provides the archetypical example.

Lear Siegler Inc provides an excellent example of the latter type: a bare bones video terminal, with a low price ($995, quantity one) and features not usually found in products of this price. This product is the ADM-3 product, which is illustrated in photo 1, with a detail of a sales message providing a sample of the text display capability in photo 2. One of the most interesting aspects of the terminal is that for those of us who are into text editing, it has an upper and lower case alphanumeric mode achieved via a unique shifted 5 by 7 dot matrix character generator. Lower case letters are generally *shifted up one* row of the matrix so that they are roughly centered. As can be seen in the example of photo 2, this gives a quite readable upper and lower case display useful in word processing applications. While it is priced a bit high for the pure hobbyist, this unit will prove attractive to the hobbyist-business set, and to commercial users of such text displays. Contact Lear Siegler Inc, Electronic Instrumentation Division, 714 N Brookhurst St, Anaheim CA, (714) 774-1010.■

All the people who bought our DUMB TERMINAL (the ADM-3) because of its low $995 unit price didn't really expect a lot. But they hadn't counted on the 32 switches. Switches that let you turn the DUMB TERMINAL into a pretty clever animal.

Take the 20 switches under the LSI name plate, for example. Among them, 11 communication rate positive action switches that let you select bauds from 192000 to 75. Also an RS232 interface extension port switch. It allows you to connect the DUMB TERMINAL to all kinds of clever devices - to recorders, printers and smarter terminals. And sitches for odd-even parity. Optional UPPER and lower case (the complete set of 128 USASCII characters)-plus a lot more._

Photo 2: Here is some text displayed on the Lear Siegler ADM-3 terminal, illustrating its unique 5 by 7 dot matrix upper and lower case display mode.

Niagara Frontier ACM

On June 17 1976, the Niagara Frontier Chapter of the Association for Computing Machinery will present a dinner meeting. BYTE's editor, Carl Helmers, will speak on "Doing It Yourself with Computers," an introductory talk on what it takes to design and build a microprocessor based general purpose computer system along with special purpose applications hardware. The talk will use electronic music as an application theme and will have a short demonstration of the concrete results of the hardware and software design it describes.

The dinner meeting will be held at the Wish-In-Well Restaurant, 8222 Main St, Clarence NY, at 6 PM. The talk is scheduled to begin at 8 PM. Reservations should be made by contacting Doug Robinson at (716) 843-7142.■

GODBOUT

BILL GODBOUT ELECTRONICS
BOX 2355, OAKLAND AIRPORT, CA 94614

FOR SALE:

We added more capacity to our warehouse. Look at these:

TANTALUM CAPACITORS
2.2 uF @ 20V..........4/$1.00
2.7 uF @ 20V..........4/$1.00
3.3 uF @ 15V..........4/$1.00
4.7 uf @ 10V..........4/$1.00
22 uF @ 10V...........3/$1.00
33 uF @ 10V...........3/$1.00
39 uF @ 10V...........3/$1.00
47 uF @ 6V............3/$1.00

ELECTROLYTIC CAPACITORS
10 uF, 250V, axial......3/$1.00
12 uF, 250V, axial......3/$1.00
100 uF, 10V, axial......5/$1.00
100 uF, 35V, PC mount....4/$1.00
100 uF, 50V, PC mount....4/$1.00
220 uF, 25V, PC mount....4/$1.00
250 uF, 25V, axial ......4/$1.00
2000 uF, 30V, PC mount...1/$0.95
4000 uF, 20V, MalloryPFP 1/$0.95
10000 uF, 10V, axial.... 1/$1.25

MYLAR CAPACITORS
Cut and formed for PC insertion. High-Q and STABLE.
.0033 uF, 50V.........10/$1.00
.005 uF, 25V..........10/$1.00
.0068 uF, 50V.........10/$1.00
.01 uF, 50V...........10/$1.00
.02 uF, 50V...........10/$1.00
.033 uF, 50V..........10/$1.00
.047 uF, 50V..........10/$1.00
.068 uF, 50V..........10/$1.00
.1 uF, 50V............10/$1.00
.22 uF, 50V...........10/$1.00
1.0 uF, 200V, 10%......3/$1.00
5.0 uF, 100V, 10%......2/$1.50
10 uF, 100V, 10%.......1/$1.50

DISC CERAMIC CAPACITORS
Small, low voltage types. Some may have leads cut and formed for PC insertion.
10 pF................10/$0.45
220 pF...............10/$0.45
.001 uF..............10/$0.50
.005 uF..............10/$0.50
.01 uF...............10/$0.50
.02 uF...............10/$0.75
.05 uF...............10/$1.00
.1 uF................10/$1.25

POLYSTYRENE CAPACITORS
Cut and formed for PC insertion. ACCURATE: 5% or better.
100 pF...............10/$1.00
150 pF...............10/$1.00
180 pF...............10/$1.00
220 pF...............10/$1.00
270 pF...............10/$1.00
390 pF...............10/$1.00
470 pF...............10/$1.00
560 pF...............10/$1.00
680 pF...............10/$1.00
820 pF...............10/$1.00
910 pF...............10/$1.00
1000 pF..............10/$1.00
1200 pF..............10/$1.00
1500 pF..............10/$1.00
1800 pF..............10/$1.00
2000 pF..............10/$1.00
2200 pF..............10/$1.00
3300 pF..............10/$1.00
3900 pF..............10/$1.00

* * * * *

WIRE WRAP SOCKETS
3 level, gold plated: use with our Hobbywrap tool, other wire wrap tools, or Wire Pencil.
14 pin...............10/$3.70
16 pin...............10/$3.85
18 pin................1/$0.75
24 pin................1/$1.00
28 pin................1/$1.25
36 pin................1/$1.35
40 pin................1/$1.75

RESISTORS 100 of any ONE value $1.70; 1000 of any ONE value $15.30. Values available:

1.0	39	1.5K	56K
1.2	47	1.8K	68K
1.5	56	2.2K	82K
1.8	68	2.7K	100K
2.2	82	3.3K	120K
2.7	100	3.9K	150K
3.3	120	4.7K	180K
3.9	150	5.6K	220K
4.7	180	6.8K	270K
5.6	220	8.2K	330K
6.8	270	10K	390K
8.2	330	12K	470K
10	390	15K	560K
12	470	18K	680K
15	560	22K	820K
18	680	27K	1.0M
22	820	33K	1.2M
27	1.0K	39K	1.5M
33	1.2K	47K	1.8M

* * * * * CAVE grafix™

SELECTED SEMICONDUCTORS
We stock too many different parts to list here---our flyer tells all, though.

DIGITAL STUFF
8093 Quad 3 state buf.....$0.63
8094 Same, but 0 gives hiZ.$0.63
8095 Noninv 3 st buf......$0.63
8096 Inv 8095 NOR enable..$0.63
8097 Noninv 4-2 3 st buf...$0.63
8098 Inv 4-2 3 state buf..$0.63
8131 6 bit buss comp hiZ in $2.50
8202 10 bit buf register...$1.10
8233 2 in 4 bit mux........$0.75
8234 2 in 4 bit mux (inv)..$0.75
8242 Quad exclusive NOR..$0.25
8250 Binary to oct decode.$0.55
8251 BCD to dec decode....$0.55
8266 2 in 4 bit mux.......$1.00
8267 2 in 4 bit mux.......$1.00
8270 4 bit shift register.$0.98
8271 4 bit shift register.$0.98
8544 Quad switch debounc.$0.90
8831 Quad 3 state drvr..$2.25
8833 Quad 3 st trscvr.....$1.90
8835 Quad 3 st trscvr.....$1.90
8837 Hex buss drv........$1.85
DS0026 Dual clock drv....$3.00
DS3608 Hex MOS/TTL cnvrt.$3.00

LINEARS
311 minidip comparator...$1.00
316 hi Z in op amp.......$2.50
318 FAST op amp..........$1.00
339 quad comparator......$1.50
340/5T plastic 5V ½A reg $1.75
(also 6, 8, 12, 15, 18, 24V) $1.75
340/5K metal 5V 1A.......$1.25
340/8K metal 8V 1A.......$1.75
340/12K metal 12V 1A.....$1.95
340/15K metal 15V 1A.....$1.85
340/18K metal 18V 1A.....$1.75
373 AM/FM/SSB IF/detect..$1.95
380M minidip 2W power amp $0.95
540 audio power driver...$1.95
565 phase locked loop....$1.00
567 tone decoder PLL.....$2.50
725 instrumentation amp..$2.00
1556M premium op amp.....$1.00
8038 VCO sine/sq/tri.....$4.50

LOW POWER SCHOTTKY
74LS00 Quad 2 in NAND....$0.40
74LS04 Hex inverter......$0.45
74LS08 Quad 2 in AND.....$0.40
74LS10 Triple 3 in NAND..$0.40
74LS20 Dual 4 in NAND....$0.40
74LS42 BCD to dec decode $1.25
74LS138 1 of 8 decode....$1.40
74LS168 Dec U/D cntr....$1.50
74LS169 Binary U/D cntr..$1.50
74LS175 Quad latch.......$1.38

COMPUTER ORIENTED KITS
ALL OUR COMPUTER KITS INCLUDE INSTRUCTIONS, DOUBLE SIDED PC BOARD, QUALITY PARTS...AND WE STAND BEHIND THEM.

"ECONORAM"..............$89.95
4K x 8 RAM board, with buffers, onboard regulation, low power, high speed, ALTAIR compatible.

"ECONOROM".............$159.95
4K x 8 EROM board for storing software. Buffers, regulation, expandable, ALTAIR compatible.

"8080 SOFTWARE BOARD"..$159.95
Same as our ECONOROM, but with editor / assembler/monitor routines pre-programmed.

"4K x 8 NAKED RAM".......$79.95
No buffers or regulation, just cost-effective memory. Compatible with JOLT systems.

"CPU POWER SUPPLY KIT"..$44.95
Designed to give power to your processor. Compatible (same size card etc.) with JOLT systems. +5V @ 5A, crowbar overvoltage protection, +12V @ ½A, -12V @ ½A, plus negative bias supply. Everything except the line cord included.

* * * * *

MICROPROCESSORS & CHIP SETS
8008 8 bit CPU..........$17.50
8080 Powerful 8 bit CPU $29.95
PACE 16 bit CPU........$125.00
8008 CHIP SET: 1-8008, 8-2102s $32.50
8080 CHIP SET: 1-8080, 8-2102s 1-5204 EROM..............$59.95
PACE CHIP SET #1: PACE IC plus all support ICs........$125.00
PACE CHIP SET #2: PACE IC plus support ICs, 32-2102, 4-5204, and PACE data packet...$195.00

MEMORY INTEGRATED CIRCUITS
1702A 2K EROM............$9.95
2102 static RAM, 750 ns..$1.95
2112 4 x 256 RAM.........$2.95
2501 256 x 1 static RAM...$1.00
5202 2K EROM.............$9.95
5203 2K EROM.............$9.95
5204 4K EROM............$17.50
5600 8 x 32 PROM..........$2.50
5610 Open col outputs....$2.95
7489 64 bit scratch pad..$2.23
93410 bipolar 256 bit RAM .$2.00

* * * * *

EROM PROGRAMMING SERVICES
We can program your 5203, 5204, or other ROM ICs for $7.50 per piece or $35 for 10 pieces. Call our 24 hr hot line to request hexadecimal coding form.

* * * * * *

REGULATED POWER SUPPLY KITS
5V, 1A..................$9.95
Stable, short proof. Add 2 lbs shipping.

HALF AMP SUPPLY KITS.....$7.95
Short proof. Specify 5, 6, 8, or 12V. Add 2 lbs shipping.

±15V SUPPLY.............$9.95
150 ma per side. Dual tracking regulator. Add 2 lbs shipping.

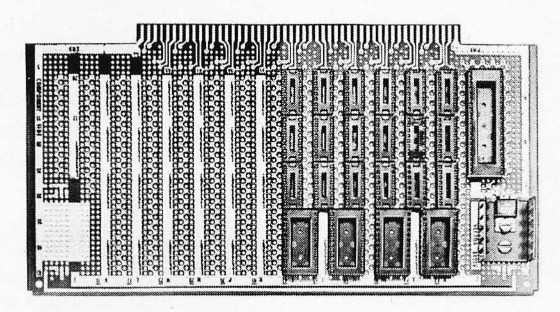
EXPERIMENTER'S SUPPLY...$10.95
Provides dual tracking, variable, regulated voltages from ±½V to over ±18V. Add 2 lbs shipping.

12V, 8A SUPPLY KIT......$22.50
New and improved. Current limits at 13 Amps; also .05V regulation or better; adjustable output 11-14V; RF proof: short circuit protected; more. Add shipping for 7 lbs.

FOR YOUR ALTAIR 8800 new from VECTOR!

Universal Prototype Board $19.95 +1 lb shp

Here is an uncommitted circuit board that plugs right into your 8800. Has Vcc & gnd lines already in, provisions for 3 regulators, and 1 heat sink included also. VECTOR is known for high quality products---this board is no exception.



HOBBYWRAP TOOL..........$41.95
Wire wrapping equipment at the right price. You get the tool (rechargeable --- no cords in tight places), bit, charger, nicads, and instructions.

PRE-PUNCHED VECTORBOARD..$8.95
Pre-punched with holes on .1" centers. 8½ x 17 inches, 1/16" thick epoxy glass base. Add 1 lb shipping per board.

* * * * *

VECTOR WIRING PENCIL.....$9.50
As reviewed in Radio-Electronics, Popular Electronics, etc. Eliminates cutting and stripping; makes interconnections between parts in 1/3 the time. Comes with tool, installed wire bobbin, extra wire bobbin, and instructions. +1lb. shp.

WIRE PENCIL REPLACEMENT WIRE
3/$2.40. Specify color choice: red, green, blue, clear.

* * * * *

TERMS: Add 50¢ to orders under $10. Add postage where indicated; otherwise items are postpaid. Bankamericard®/Mastercharge® call (415) 357-7007, 24 hours a day. Californians add tax.

OUR FLYER HAS THE STORY ON LOTS OF OTHER PARTS AND KITS, AS WELL AS FLASHY ARTWORK, AN ORDER BLANK YOU CAN CUT OUT AND SEND IN, VARIOUS PICTURES, GOBS OF DIFFERENT TYPEFACES, BORDERS, HALFTONE ARTWORK, AND MOST IMPORTANT OF ALL...COMPETITIVE PRICING.
SEND FOR OUR FLYER...SEND FOR OUR FLYER...SEND FOR OUR FLYER...

What's New?

A Complete Desk Top Assembly Language System

For individuals and companies requiring a completely integrated software development system, one place to look is the capital goods market. An excellent example of such a product, which is well below the price of many existing microcomputer development systems, is this desk top computer from Tranti Systems, the uScope Model 8000 programming system. The uScope 8000 is unusual in that it is the programming devel-

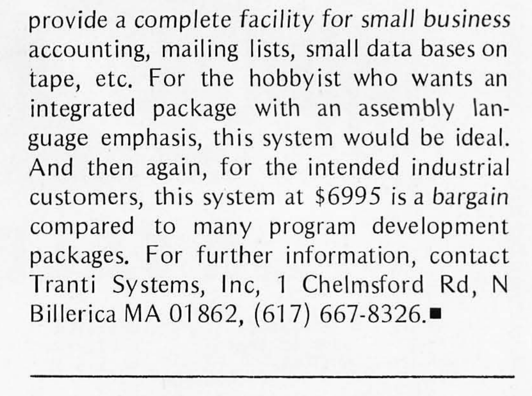


opment system equivalent of one of the desk top BASIC oriented packages available from several manufacturers. However, instead of generating high level language interpretive code, it has an excellent interactive assembler so that optimized code can be crafted by hand. (According to Frank Trantanella, president, a high level language software package is presently being explored for even greater program generation efficiency.) As a complete computer system, the desk top unit contains:

- Resident ROM software for memory editor, interactive assembler, monitor
- ASCII text keyboard
- Numeric and special function keyboard
- Video display
- Alphanumeric printer
- Tape cartridge mass storage

The standard configuration has an 8080 processor, with an 8080 assembly code data table. However, since the assembler is written in a generalized fashion, the same system hardware can be used to generate (but not execute or test) code for virtually any 8 bit microprocessor architecture. This change is accomplished by reading in an alternate data table from tape.

For the businessman hobbyist who wants to learn and understand assembly language applications of the 8080, this system could



provide a complete facility for small business accounting, mailing lists, small data bases on tape, etc. For the hobbyist who wants an integrated package with an assembly language emphasis, this system would be ideal. And then again, for the intended industrial customers, this system at $6995 is a bargain compared to many program development packages. For further information, contact Tranti Systems, Inc, 1 Chelmsford Rd, N Billerica MA 01862, (617) 667-8326.■

An 8080 Operating System

A micro operating system for 8080 based microcomputers is now available from its originator D S Marcus. Its two main features are a monitor and a system call facility. The monitor provides a complete set of debugging aids and a program load capability. System calls allow a user program to invoke the operating system for performing various input/ouput related functions.

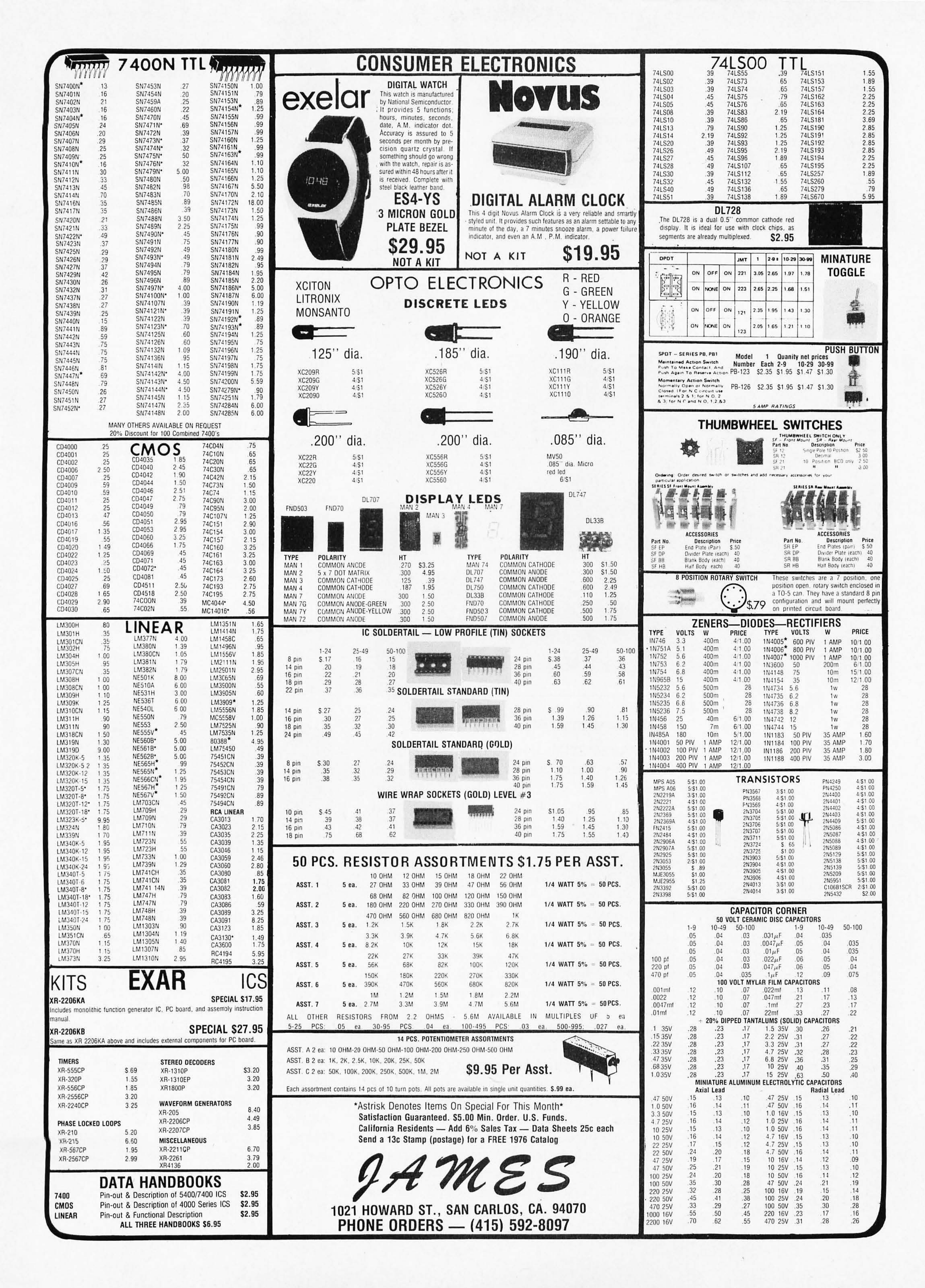
Communication with the monitor is done via a command language. Individual commands are provided for inspecting and modifying both memory and 8080 user registers, setting breakpoints, masked memory searching, initializing a block of memory, dumping memory (both numeric and ASCII conversion provided), and more. Command operands can be given as arbitrary expressions consisting of numeric terms (octal, decimal and hexadecimal), ASCII and register value terms. The conversion mode for values typed by the monitor can be set to octal, decimal or hexadecimal.

System calls are provided for doing both character and line oriented input/output. Provision is made for supporting various user terminals including Teletype compatible devices. In addition, the operating system is written in such a way that it may reside in ROM (read only memory).

A minimum memory configuration of 4 K bytes is required to run the operating system. It is supplied as a proprietary software package on a self loading binary paper tape with a complete user's manual. It may be obtained for $50 from D S Marcus, PO Box 17066, Irvine CA 92713.■

Free Op Code Table Reprint

Stanley Veit, storekeeper of the Computer Mart of New York, Inc, 314 Fifth Av, New York NY 10001, will send you a free copy of a Motorola M6800 operation codes table and reference sheet. Send him a self addressed stamped envelope.

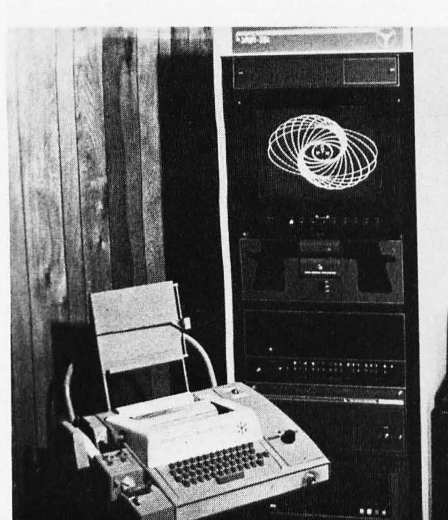


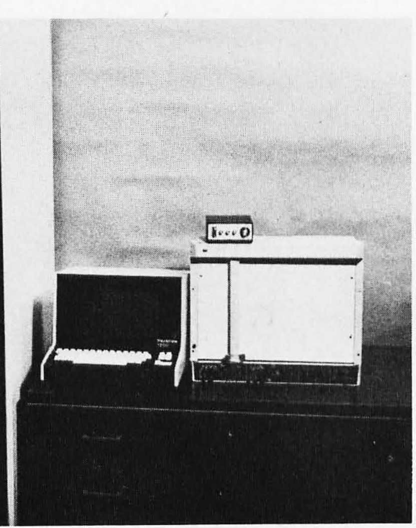
7400N TTL
SN7400N* .13 SN7453N .27 SN74150N 1.00
SN7401N .16 SN7454N .20 SN74151N .79
SN7402N .21 SN7459A .25 SN74153N .89
SN7403N .16 SN7460N .22 SN74154N* 1.25
SN7404N* .16 SN7470N .45 SN74155N .99
SN7405N .24 SN7471N* .69 SN74156N .99
SN7406N .20 SN7472N .39 SN74157N .99
SN7407N .29 SN7473N* .37 SN74160N 1.25
SN7408N .25 SN7474N* .32 SN74161N .99
SN7409N .25 SN7475N* .50 SN74163N* .99
SN7410N* .16 SN7476N* .32 SN74164N 1.10
SN7411N .30 SN7479N* 5.00 SN74165N 1.10
SN7412N .33 SN7480N .50 SN74166N 1.25
SN7413N .45 SN7482N .98 SN74167N 5.50
SN7414N .70 SN7483N .70 SN74170N 2.10
SN7416N .35 SN7485N .89 SN74172N 18.00
SN7417N .35 SN7486N .39 SN74173N 1.50
SN7420N .21 SN7488N 3.50 SN74174N 1.25
SN7421N .33 SN7489N 2.25 SN74175N .99
SN7422N* .49 SN7490N* .45 SN74176N .90
SN7423N .37 SN7491N .75 SN74177N .90
SN7425N .29 SN7492N .49 SN74180N .99
SN7426N .29 SN7493N* .49 SN74181N 2.49
SN7427N .37 SN7494N .79 SN74182N .95
SN7429N .42 SN7495N .79 SN74184N 1.95
SN7430N .26 SN7496N .89 SN74185N 2.20
SN7432N .31 SN7497N* 4.00 SN74186N* 5.00
SN7437N .27 SN74100N* 1.00 SN74187N 6.00
SN7438N .27 SN74107N .39 SN74190N 1.19
SN7439N .25 SN74121N* .39 SN74191N 1.25
SN7440N .15 SN74122N .39 SN74192N* .89
SN7441N .89 SN74123N* .70 SN74193N* .89
SN7442N .59 SN74125N .60 SN74194N 1.25
SN7443N .75 SN74126N .60 SN74195N .75
SN7444N .75 SN74132N 1.09 SN74196N 1.25
SN7445N .75 SN74136N .95 SN74197N .75
SN7446N .81 SN74141N 1.15 SN74198N 1.75
SN7447N* .69 SN74142N* 4.00 SN74199N 1.75
SN7448N .79 SN74143N* 4.50 SN74200N 5.59
SN7450N .26 SN74144N* 4.50 SN74279N* .90
SN7451N .27 SN74145N 1.15 SN74251N 1.79
SN7452N* .27 SN74147N 2.35 SN74284N 6.00
SN74148N 2.00 SN74285N 6.00
MANY OTHERS AVAILABLE ON REQUEST
20% Discount for 100 Combined 7400's
CMOS
CD4000 .25 CD4035 1.85 74C04N .75
CD4001 .25 CD4040 2.45 74C10N .65
CD4002 .25 CD4042 1.90 74C20N .65
CD4006 2.50 CD4044 1.50 74C30N .65
CD4007 .25 CD4046 2.51 74C42N 2.15
CD4009 .59 CD4047 2.75 74C73N 1.50
CD4010 .59 CD4049 .79 74C74 1.15
CD4011 .25 CD4050 .79 74C90N 3.00
CD4012 .25 CD4051 2.95 74C95N 2.00
CD4013 .47 CD4053 2.95 74C107N 1.25
CD4016 .56 CD4060 3.25 74C151 2.90
CD4017 1.35 CD4066 1.75 74C154 3.00
CD4019 .55 CD4069 .45 74C157 2.15
CD4020 1.49 CD4071 .45 74C160 3.25
CD4022 1.25 CD4072* .45 74C161 3.25
CD4023 .25 CD4081 .45 74C163 3.00
CD4024 1.50 CD4511 2.50 74C164 3.25
CD4025 .25 CD4518 2.50 74C173 2.60
CD4027 .69 74C00N .39 74C193 2.75
CD4028 1.65 74C02N .55 74C195 2.75
CD4029 2.90 MC4044* 4.50
CD4030 .65 MC14016* .56
LINEAR
LM300H .80 LM377N 4.00 LM1351N 1.65
LM301H .35 LM380N 1.39 LM1414N 1.75
LM301CN .35 LM380CN 1.05 LM1458C .65
LM302H .75 LM381N 1.79 LM1496N .95
LM304H 1.00 LM382N 1.79 LM1556V 1.85
LM305H .95 NE501K 8.00 LM2111N 1.95
LM307CN .35 NE510A 6.00 LM2901N 2.95
LM308H 1.00 NE531H 3.00 LM3065N .69
LM308CN 1.00 NE536T 6.00 LM3900N .55
LM309H 1.10 NE540L 6.00 LM3905N .60
LM309K 1.25 NE550N .79 LM3909* 1.25
LM310CN 1.15 NE553 2.50 LM5556N 1.85
LM311H .90 NE555V* .45 MC5558V 1.00
LM311N .90 NE560B* 5.00 LM7525N .90
LM318CN 1.50 NE561B* 5.00 LM7535N 1.25
LM319N 1.30 NE562B* 5.00 80388* 4.95
LM319D 9.00 NE565H* 1.95 LM75450 .49
LM320K-5 1.35 NE565N* 1.25 75451CN .39
LM320K-5.2 1.35 NE566CN* 1.95 75452CN .39
LM320K-12 1.35 NE567H* 1.25 75453CN .39
LM320K-15 1.35 NE567V* 1.50 75454CN .39
LM320T-5* 1.75 LM703CN .45 75491CN .79
LM320T-8* 1.75 LM709H .29 75492CN .89
LM320T-12* 1.75 LM709N .29 75494CN .89
LM320T-18* 1.75 LM710N .79 RCA LINEAR
LM323K-5* 9.95 LM711N .39 CA3013 1.70
LM324N 1.80 LM723N .55 CA3023 2.15
LM339N 1.70 LM723H .55 CA3035 2.25
LM340K-5 1.95 LM733N 1.00 CA3039 1.35
LM340K-12 1.95 LM739N 1.29 CA3046 1.15
LM340K-15 1.95 LM741CH .35 CA3059 2.46
LM340K-24 1.95 LM741CN .35 CA3060 2.80
LM340T-5 1.75 LM741 14N .39 CA3080 .85
LM340T-6 1.75 LM747H .79 CA3081 1.75
LM340T-8* 1.75 LM747N .79 CA3082 2.00
LM340T-18* 1.75 LM748H .39 CA3083 1.60
LM340T-12 1.75 LM748N .39 CA3086 .59
LM340T-15 1.75 LM1303N .90 CA3089 3.25
LM340T-24 1.75 LM1304N 1.19 CA3091 8.25
LM350N 1.00 LM1305N 1.40 CA3123 1.85
LM351CN .65 LM1307N .85 CA3130* 1.49
LM370N 1.15 LM1310N 2.95 CA3600 1.75
LM370H 1.15 RC4194 5.95
LM373N 3.25 RC4195 3.25
KITS EXAR ICS
XR-2206KA SPECIAL $17.95
Includes monolithic function generator IC, PC board, and assembly instruction manual.
XR-2206KB SPECIAL $27.95
Same as XR 2206KA above and includes external components for PC board.
TIMERS
XR-555CP $.69
XR-320P 1.55
XR-556CP 1.85
XR-2556CP 3.20
XR-2240CP 3.25
PHASE LOCKED LOOPS
XR-210 5.20
XR-215 6.60
XR-567CP 1.95
XR-2567CP 2.99
STEREO DECODERS
XR-1310P $3.20
XR-1310EP 3.20
XR1800P 3.20
WAVEFORM GENERATORS
XR-205 8.40
XR-2206CP 4.49
XR-2207CP 3.85
MISCELLANEOUS
XR-2211CP 6.70
XR-2261 3.79
XR4136 2.00
DATA HANDBOOKS
7400 Pin-out & Description of 5400/7400 ICS $2.95
CMOS Pin-out & Description of 4000 Series ICS $2.95
LINEAR Pin-out & Functional Description $2.95
ALL THREE HANDBOOKS $6.95
CONSUMER ELECTRONICS
exelar
DIGITAL WATCH
This watch is manufactured by National Semiconductor. It provides 5 functions; hours, minutes, seconds, date, A.M. indicator dot. Accuracy is assured to 5 seconds per month by precision quartz crystal. If something should go wrong with the watch, repair is assured within 48 hours after it is received. Complete with steel black leather band.
ES4-YS
3 MICRON GOLD PLATE BEZEL
$29.95
NOT A KIT
Novus
DIGITAL ALARM CLOCK
This 4 digit Novus Alarm Clock is a very reliable and smartly styled unit. It provides such features as an alarm settable to any minute of the day, a 7 minutes snooze alarm, a power failure indicator, and even an A.M., P.M. indicator.
NOT A KIT $19.95
OPTO ELECTRONICS
XCITON
LITRONIX
MONSANTO
DISCRETE LEDS
R - RED
G - GREEN
Y - YELLOW
O - ORANGE
.125" dia.
XC209R 5/$1
XC209G 4/$1
XC209Y 4/$1
XC2090 4/$1
.185" dia.
XC526R 5/$1
XC526G 4/$1
XC526Y 4/$1
XC5260 4/$1
.190" dia.
XC111R 5/$1
XC111G 4/$1
XC111Y 4/$1
XC1110 4/$1
.200" dia.
XC22R 5/$1
XC22G 4/$1
XC22Y 4/$1
XC220 4/$1
.200" dia.
XC556R 5/$1
XC556G 4/$1
XC556Y 4/$1
XC5560 4/$1
.085" dia.
MV50
.085" dia. Micro red led
6/$1
DISPLAY LEDS
FND503 FND70 DL707 MAN 2 MAN 3 MAN 4 MAN 7 DL747 DL33B
TYPE POLARITY HT
MAN 1 COMMON ANODE .270 $3.25
MAN 2 5 x 7 DOT MATRIX .300 4.95
MAN 3 COMMON CATHODE .125 .39
MAN 4 COMMON CATHODE .187 1.95
MAN 7 COMMON ANODE .300 1.50
MAN 7G COMMON ANODE-GREEN .300 2.50
MAN 7Y COMMON ANODE-YELLOW .300 2.50
MAN 72 COMMON ANODE .300 1.50
MAN 74 COMMON CATHODE .300 $1.50
DL707 COMMON ANODE .300 $1.50
DL747 COMMON ANODE .600 2.25
DL750 COMMON CATHODE .600 2.49
DL33B COMMON CATHODE .110 1.25
FND70 COMMON CATHODE .250 .50
FND503 COMMON CATHODE .500 1.75
FND507 COMMON ANODE .500 1.75
IC SOLDERTAIL — LOW PROFILE (TIN) SOCKETS
1-24 25-49 50-100
8 pin $.17 .16 .15
14 pin .20 .19 .18
16 pin .22 .21 .20
18 pin .29 .28 .27
22 pin .37 .36 .35
24 pin $.38 .37 .36
28 pin .45 .44 .43
36 pin .60 .59 .58
40 pin .63 .62 .61
SOLDERTAIL STANDARD (TIN)
14 pin $.27 .25 .24
16 pin .30 .27 .25
18 pin .35 .32 .30
24 pin .49 .45 .42
28 pin $.99 .90 .81
36 pin 1.39 1.26 1.15
40 pin 1.59 1.45 1.30
SOLDERTAIL STANDARD (GOLD)
8 pin $.30 .27 .24
14 pin .35 .32 .29
16 pin .38 .35 .32
24 pin $.70 .63 .57
28 pin 1.10 1.00 .90
36 pin 1.75 1.40 1.26
40 pin 1.75 1.59 1.45
WIRE WRAP SOCKETS (GOLD) LEVEL #3
10 pin $.45 .41 .37
14 pin .39 .38 .37
16 pin .43 .42 .41
18 pin .75 .68 .62
24 pin $1.05 .95 .85
28 pin 1.40 1.25 1.10
36 pin 1.59 1.45 1.30
40 pin 1.75 1.55 1.40
50 PCS. RESISTOR ASSORTMENTS $1.75 PER ASST.
ASST. 1 5 ea. 10 OHM 12 OHM 15 OHM 18 OHM 22 OHM 27 OHM 33 OHM 39 OHM 47 OHM 56 OHM 1/4 WATT 5% = 50 PCS.
ASST. 2 5 ea. 68 OHM 82 OHM 100 OHM 120 OHM 150 OHM 180 OHM 220 OHM 270 OHM 330 OHM 390 OHM 1/4 WATT 5% = 50 PCS.
ASST. 3 5 ea. 470 OHM 560 OHM 680 OHM 820 OHM 1K 1.2K 1.5K 1.8K 2.2K 2.7K 1/4 WATT 5% = 50 PCS.
ASST. 4 5 ea. 3.3K 3.9K 4.7K 5.6K 6.8K 8.2K 10K 12K 15K 18K 1/4 WATT 5% = 50 PCS.
ASST. 5 5 ea. 22K 27K 33K 39K 47K 56K 68K 82K 100K 120K 1/4 WATT 5% = 50 PCS.
ASST. 6 5 ea. 150K 180K 220K 270K 330K 390K 470K 560K 680K 820K 1/4 WATT 5% = 50 PCS.
ASST. 7 5 ea. 1M 1.2M 1.5M 1.8M 2.2M 2.7M 3.3M 3.9M 4.7M 5.6M 1/4 WATT 5% = 50 PCS.
ALL OTHER RESISTORS FROM 2.2 OHMS - 5.6M AVAILABLE IN MULTIPLES OF 5 ea 5-25 PCS: .05 ea. 30-95 PCS: .04 ea. 100-495 PCS: .03 ea. 500-995: .027 ea.
14 PCS. POTENTIOMETER ASSORTMENTS
ASST. A 2 ea: 10 OHM-20 OHM-50 OHM-100 OHM-200 OHM-250 OHM-500 OHM
ASST. B 2 ea: 1K, 2K, 2.5K, 10K, 20K, 25K, 50K
ASST. C 2 ea: 50K, 100K, 200K, 250K, 500K, 1M, 2M
$9.95 Per Asst.
Each assortment contains 14 pcs of 10 turn pots. All pots are available in single unit quantities. $.99 ea.
Astrisk Denotes Items On Special For This Month
Satisfaction Guaranteed. $5.00 Min. Order. U.S. Funds.
California Residents — Add 6% Sales Tax — Data Sheets 25c each
Send a 13c Stamp (postage) for a FREE 1976 Catalog
JAMES
1021 HOWARD ST., SAN CARLOS, CA. 94070
PHONE ORDERS — (415) 592-8097
74LS00 TTL
74LS00 .39 74LS55 .39 74LS151 1.55
74LS02 .39 74LS73 .65 74LS153 1.89
74LS03 .39 74LS74 .65 74LS157 1.55
74LS04 .45 74LS75 .79 74LS162 2.25
74LS05 .45 74LS76 .65 74LS163 2.25
74LS08 .39 74LS83 2.19 74LS164 2.25
74LS10 .39 74LS86 .65 74LS181 3.69
74LS13 .79 74LS90 1.25 74LS190 2.85
74LS14 2.19 74LS92 1.25 74LS191 2.85
74LS20 .39 74LS93 1.25 74LS192 2.85
74LS26 .49 74LS95 2.19 74LS193 2.85
74LS27 .45 74LS96 1.89 74LS194 2.25
74LS28 .49 74LS107 .65 74LS195 2.25
74LS30 .39 74LS112 .65 74LS257 1.89
74LS32 .45 74LS132 1.55 74LS260 .55
74LS40 .49 74LS136 .65 74LS279 .79
74LS51 .39 74LS138 1.89 74LS670 5.95
DL728
The DL728 is a dual 0.5" common cathode red display. It is ideal for use with clock chips, as segments are already multiplexed. $2.95
MINATURE TOGGLE
DPDT JMT 1 2-9 10-29 30-99
ON OFF ON 221 3.05 2.65 1.97 1.78
ON NONE ON 223 2.65 2.25 1.68 1.51
ON OFF ON 121 2.35 1.95 1.43 1.30
ON NONE ON 123 2.05 1.65 1.21 1.10
PUSH BUTTON
SPDT — SERIES PB, PB1
Model Number 1 Each Quanity net prices 2-9 10-29 30-99
Maintained Action Switch Push To Make Contact And Push Again To Reserve Action PB-123 $2.35 $1.95 $1.47 $1.30
Momentary Action Switch Normally Open or Normally Closed (For N C circuit use terminals 2 & 1; for N O 2 & 3; for N C and N O 1,2 &3) PB-126 $2.35 $1.95 $1.47 $1.30
5 AMP RATINGS
THUMBWHEEL SWITCHES
THUMBWHEEL SWITCH ONLY
SF — Front Mount SR — Rear Mount
Part No Description Price
SF 12 Single Pole 10 Position $2.50
SR 12 Decimal 3.00
SF 21 10 Position BCD only 2.50
SR 21 " " 3.00
Ordering: Order desired switch or switches and add necessary accessories for your particular application.
SERIES SF Front Mount Assembly
SERIES SR Rear Mount Assembly
ACCESSORIES
Part No. Description Price
SF EP End Plate (Pair) $.50
SF DP Divider Plate (each) .40
SF BB Blank Body (each) .40
SF HB Half Body (each) .40
ACCESSORIES
Part No. Description Price
SR EP End Plates (pair) $.50
SR DP Divider Plate (each) .40
SR BB Blank Body (each) .40
SR HB Half Body (each) .40
8 POSITION ROTARY SWITCH
$.79
These switches are a 7 position, one position open, rotary switch enclosed in a TO-5 can. They have a standard 8 pin configuration and will mount perfectly on printed circuit board.
ZENERS—DIODES—RECTIFIERS
TYPE VOLTS W PRICE TYPE VOLTS W PRICE
1N746 3.3 400m 4/1.00 1N4005* 600 PIV 1 AMP 10/1.00
1N751A 5.1 400m 4/1.00 1N4006* 800 PIV 1 AMP 10/1.00
1N752 5.6 400m 4/1.00 1N4007* 1000 PIV 1 AMP 10/1.00
1N753 6.2 400m 4/1.00 1N3600 50 200m 6/1.00
1N754 6.8 400m 4/1.00 1N4148 75 10m 15/1.00
1N965B 15 400m 4/1.00 1N4154 35 10m 12/1.00
1N5232 5.6 500m .28 1N4734 5.6 1w .28
1N5234 6.2 500m .28 1N4735 6.2 1w .28
1N5235 6.8 500m .28 1N4736 6.8 1w .28
1N5236 7.5 500m .28 1N4738 8.2 1w .28
1N456 25 40m 6/1.00 1N4742 12 1w .28
1N458 150 7m 6/1.00 1N4744 15 1w .28
1N485A 180 10m 5/1.00 1N1183 50 PIV 35 AMP 1.60
1N4001 50 PIV 1 AMP 12/1.00 1N1184 100 PIV 35 AMP 1.70
1N4002 100 PIV 1 AMP 12/1.00 1N1186 200 PIV 35 AMP 1.80
1N4003 200 PIV 1 AMP 12/1.00 1N1188 400 PIV 35 AMP 3.00
1N4004 400 PIV 1 AMP 12/1.00
TRANSISTORS
MPS A05 5/$1.00
MPS A06 5/$1.00
2N2219A 3/$1.00
2N2221 4/$1.00
2N2222A 5/$1.00
2N2369 5/$1.00
2N2369A 4/$1.00
FN2415 5/$1.00
2N2484 4/$1.00
2N2906A 4/$1.00
2N2907A 5/$1.00
2N2925 5/$1.00
2N3053 2/$1.00
2N3055 $.89
MJE3055 $1.00
MJE2955 $1.25
2N3392 5/$1.00
2N3398 5/$1.00
PN3567 3/$1.00
PN3568 4/$1.00
PN3569 4/$1.00
2N3704 5/$1.00
2N3705 5/$1.00
2N3706 5/$1.00
2N3707 5/$1.00
2N3711 5/$1.00
2N3724 $.65
2N3725 $1.00
2N3903 5/$1.00
2N3904 4/$1.00
2N3905 4/$1.00
2N3906 4/$1.00
2N4013 3/$1.00
2N4014 3/$1.00
PN4249 4/$1.00
PN4250 4/$1.00
2N4400 4/$1.00
2N4401 4/$1.00
2N4402 4/$1.00
2N4403 4/$1.00
2N4409 5/$1.00
2N5086 4/$1.00
2N5087 4/$1.00
2N5088 4/$1.00
2N5089 4/$1.00
2N5129 5/$1.00
2N5138 5/$1.00
2N5139 5/$1.00
2N5209 5/$1.00
2N5951 5/$1.00
C106B1SCR 2/$1.00
2N5432 $2.00
CAPACITOR CORNER
50 VOLT CERAMIC DISC CAPACITORS
1-9 10-49 50-100 1-9 10-49 50-100
10 pf .05 .04 .03 .001µF .05 .04 .035
22 pf .05 .04 .03 .0047µF .05 .04 .035
47 pf .05 .04 .03 .01µF .05 .04 .035
100 pf .05 .04 .03 .022µF .06 .05 .04
220 pf .05 .04 .03 .047µF .06 .05 .04
470 pf .05 .04 .035 .1µF .12 .09 .075
100 VOLT MYLAR FILM CAPACITORS
.001mf .12 .10 .07 .022mf .13 .11 .08
.0022 .12 .10 .07 .047mf .21 .17 .13
.0047mf .12 .10 .07 .1mf .27 .23 .17
.01mf .12 .10 .07 .22mf .33 .27 .22
+ 20% DIPPED TANTALUMS (SOLID) CAPACITORS
.1 35V .28 .23 .17 1.5 35V .30 .26 .21
.15 35V .28 .23 .17 2.2 25V .31 .27 .22
.22 35V .28 .23 .17 3.3 25V .31 .27 .22
.33 35V .28 .23 .17 4.7 25V .32 .28 .23
.47 35V .28 .23 .17 6.8 25V .36 .31 .25
.68 35V .28 .23 .17 10 25V .40 .35 .29
1.0 35V .28 .23 .17 15 25V .63 .50 .40
MINIATURE ALUMINUM ELECTROLYTIC CAPACITORS
Axial Lead Radial Lead
.47 50V .15 .13 .10 .47 25V .15 .13 .10
1.0 50V .16 .14 .11 .47 50V .16 .14 .11
3.3 50V .15 .13 .10 1.0 16V .15 .13 .10
4.7 25V .16 .14 .12 1.0 25V .16 .14 .11
10 25V .15 .13 .10 1.0 50V .16 .14 .11
10 50V .16 .14 .12 4.7 16V .15 .13 .10
22 25V .17 .15 .12 4.7 25V .15 .13 .10
22 50V .24 .20 .18 4.7 50V .16 .14 .11
47 25V .19 .17 .15 10 16V .14 .12 .09
47 50V .25 .21 .19 10 25V .15 .13 .10
100 25V .24 .20 .18 10 50V .16 .14 .12
100 50V .35 .30 .28 47 50V .24 .21 .19
220 25V .32 .28 .25 100 16V .19 .15 .14
220 50V .45 .41 .38 100 25V .24 .20 .18
470 25V .33 .29 .27 100 50V .35 .30 .28
1000 16V .55 .50 .45 220 16V .23 .17 .16
2200 16V .70 .62 .55 470 25V .31 .28 .26

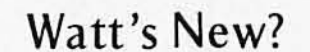
What's New?

Space War, Anyone?

The deluxe way to accomplish sophisticated games and activities involving graphics is of course to start or go to work for a company which has such computerized graphics as a way of life. An example of a type of system you're likely to find in such a context is this Scientific Process & Research Inc SPAR/GRAPHICS system based on a Data General NOVA 3 or ECLIPSE minicomputer, hard surface disk drive, 1024 by 1024 point display, CRT terminal and plotter. The hardware of course is not all that's involved in such a system, and SPAR provides a powerful graphics command language to go along with the unit. The software is documented to the level of standard FORTRAN calls so that user developed FORTRAN programs can also manipulate the display. The software handles *two and* three dimensional images, objects in motion, rotations, etc. Systems are available from SPAR starting at $5000 minus the cost of the NOVA 3 or ECLIPSE disk system. Descriptive literature is offered on letterhead request. Contact Scientific Process & Research Inc, 24 N Third Av, Highland Park NJ 08904.■



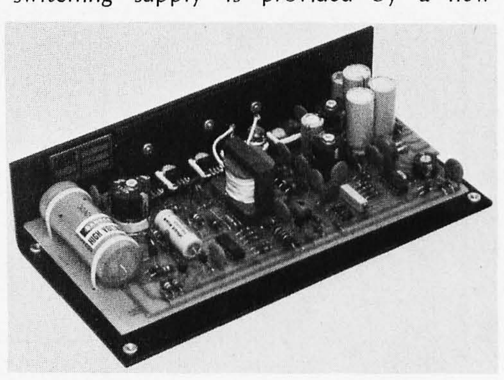




Watt's New?

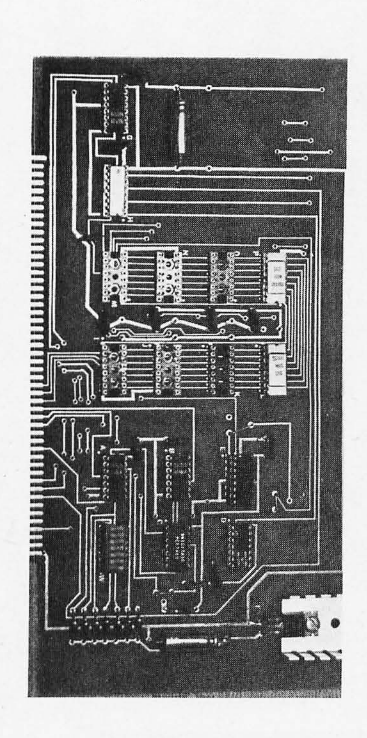
One of the biggest drawbacks of most power supplies (check the surplus advertisements) is what might be called the "power to weight ratio." A typical conventional power supply of 5 V at 10.5 A (52.5 W) might weigh 16 pounds, a large amount of iron to lug around, especially if you want to make your system portable. One of the technological solutions to this excessive weight is a form of power supply which uses what is called the "switching regulator" concept.

An example of the weight savings of a switching supply is provided by a new product from Boschert Associates, 1031 E Duane, Suite C, Sunnyvale CA 94086. This product is the model OL80 power supply, which weighs a mere two pounds and provides +5 V at 10 A, +12 V at 2 A, −12 V at 2 A, and either −9 V or −5 V at 2 A. The design has overvoltage, overcurrent and reverse voltage protection. The total power output is 80 W maximum. Just looking at the 5 V supply alone, the 50 W logic capability of this switching supply gives a power to weight ratio of 25 as compared to the randomly picked conventional supply's power to weight ratio of 3.2. In small quantities, this power supply costs $199; however, it should prove more than adequate for many microcomputer applications with its large logic power capacity and ample capacity for linear interfaces. ■



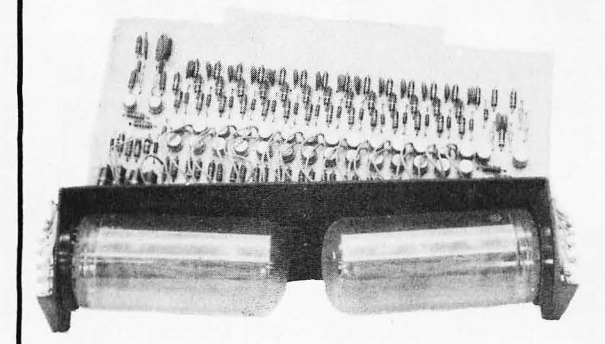
Fill Space With ROMs

An Altair 8800 compatible Programmable Read Only Memory card which takes advantage of the very high speed, low cost, plentiful supply, and ease of programming of bipolar PROMs is available from Digitech, PO Box 6838, Grosse Pointe MI 48236. The card may contain a maximum of either 1 K or 2 K of 8 bit bytes using either industry standard 256 x 4 or 512 x 4 bipolar fusible link PROMs, respectively. A board cycle time of 180 ns maximum allows the 8080 processor to run at maximum speed with no waiting for memory data. Full provision is made on the board for power regulation, filtering, and decoupling, address and control signal decoding, and data output buffering. The address for the board is very simply set by a DIP switch mounted on the board. All circuits are socket mounted with the exception of the +5 V regulator which is mounted on a heatsink. The board itself is type G-10 epoxy and is double sided with plated through holes. The decoding circuits are low power 74LS series. The board comes completely assembled and tested, minus PROMs.■



DELTA ELECTRONICS CO

POST OFFICE BOX 2, AMESBURY, MASS. 01913 Phone (617) 388-4705



Giant Alpha—Numeric NIXIE Tubes

One of the ultimate forms of computer output. These Burroughs NIXIE tubes have characters 2½" high, easily readable from long distances. The character is a 15 segment Alpha-Numeric type. The tubes (no. B7971) are mounted in 2 sockets on a PC board, which contains 33 driver transistors. We supply data on the tubes. Ideal for large clocks, store displays, sports scoreboards, or any other alpha-numeric display. Shipping weight 2 lbs.

STOCK NO. B6001 set of 2 tubes, sockets, PC board $6.95, 2 sets/12.00

Laboratory Regulated Power Supply



This power supply was designed for use in a small computer system where performance and reliability were an absolute must. 5 output voltages are available:

1. 12 volts DC @ 1.0 amp
2. 14 volts DC @ 2.9 amps
3. 18 volts DC @ 6.6 amps
4. 26.5 volts DC @ 3.25 amps
5. 24 volts DC @ 1.6 amp

All voltages are semiconductor rectified and highly filtered, and may be run at full load at the same time. Each output is fused seperately, and the entire supply is circuit breaker protected. A switched AC outlet is available on the front panel. This would be an ideal supply for a small system; each board could have its own on-board voltage regulators. Also good as a general purpose laboratory power supply. Use it stand-alone or rack mounted. 17½" wide x 5" deep x 6¼" high. Shipping weight 35 lbs.

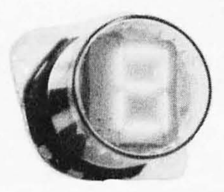
STOCK NO. B5025 $27.50 each, 2 for $50

5 volt Power Supply Kit

This power supply kit contains all the parts needed to construct a 5.0 volt regulated power supply, rated at 1 amp. Ideal for small DTL or TTL projects, and as a breadboard power supply. The kit contains the following: power transformer, LM309 5v regulator, TO-3 socket for the LM309, line cord, pilot light, power switch, and a printed circuit board containing a bridge rectifier, fuse holder, filter capacitor, and bleeder resistor. Data supplied. Shipping weight 3 lbs.

STOCK NO. B5097 Complete 5 volt power supply kit $9.50 each, 3 kits for $25

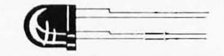
RCA End View NUMITRONS



The RCA end view NUMITRON is a 7 segment incandescent readout tube, with a character height of 5/8". It requires a 9 pin Novar socket, which we furnish with each tube. The tube operates at a nominal voltage of 4.5v, and draws 24 ma. per segment. It can be filtered to any color. With data.

STOCK NO. B5207 with sockets $2.75 ea, 4/10.00, 8/18.00

OPCOA RED LEDs



OPCOA no. OSL-3-30 red LED. All are new & tested good. Use in all types of displays. .20" dia., can be panel or PC board mounted.

STOCK NO. B4536 .25 each, 5/$1.00

DIODE-LITE™ LED LOGIC/STATE FAULT INDICATORS



DIALIGHT 555-2003 GaAs LED indicator, red diffused light. Built in series resistor to make it directly usable with TTL; draws 6 ma. @ 5 volts. .10" wide x .24" deep x .25" high. Vert, PC mount.

STOCK NO. B4537 .50 each, 5/$2.00

NiCad Batteries



Nickel Cadmium rechargeable batteries (NiCads) are highly desirable items, useful in many applications. Although their initial cost might seem high, the fact that they can be recharged many hundreds of times makes the per use cost negligable. We have a good selection of sealed NiCads made by Burgess. Our low prices are about 50% of current distributor prices. NiCads are very useful in keeping memory up in case of power failure. Many other uses of course.

Burgess no.	Volts	Size	Stock no.	Price
.600 SC *	1.2	.600 x 1.95	B5125	$1.50 ea, 4/5.50
CD13L	1.2	1.27 x 1.44	B5289	3.25 ea, 4/12.00
CD22	6.0	1.06 x 1.97	B5290	4.25 ea, 4/16.00
CD24	9.6	1.06 x 2.65	B5291	6.75 ea, 4/25.00
CD25	9.6	1.06 x 3.20	B5292	6.95 ea, 4/26.00
CD27L	12	1.065 x 2.93	B5293	7.50 ea, 4/28.00

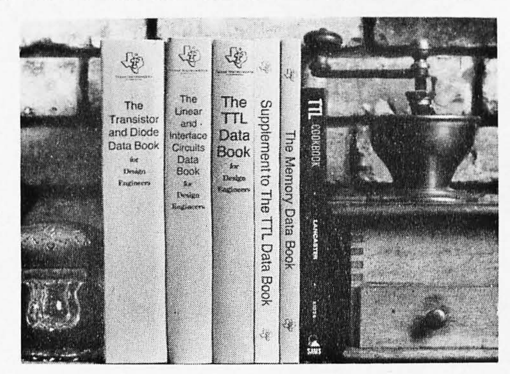
* Gould brand

Heat Sink

A small aluminum heat sink for TO-220, etc. 1¼" x 1 1/8" x ½" high. Could be mounted on PC board for regulators or power transistors.

STOCK NO. B7192 5/$1.00

Send for our latest free catalog. We welcome Mastercharge & BankAmeriCard orders; we must have ALL the numbers on the card for processing. Please include sufficient postage (2 lbs. min.); excess will be refunded. Minimum order $5.



You'll Want to Nybble at these Byte Books

Where does the editor of a computer magazine turn to when he must verify some author's hardware design? Information on a 75450 interface gate, or a 74147 priority encoder circuit does not spring forth by magic. Checking the information supplied by authors is part of BYTE's quality control program.

When you build a project, you need this same sort of information. All you find in the advertisements for parts are mysterious numbers identifying the little beasties . . . hardly the sort of information which can be used to design a custom logic circuit. You can find out about many of the numbers by using the information found in these books. No laboratory bench is complete without an accompanying library shelf filled with references – and this set of Texas Instruments engineering manuals plus Don Lancaster's *TTL Cookbook* will provide an excellent starting point or addition to your personal library.

● **The TTL Cookbook** by Don Lancaster, published by Howard W. Sams, Indianapolis, Indiana. Start your quest for data here with Don's tutorial explanations of what makes a TTL logic design tick. 335 pages, $8.95 postpaid.

● **The TTL Data Book for Design Engineers,** by Texas Instruments Incorporated. How does an engineer find out about the TTL circuits? He reads the manufacturer's literature. This 640 page beauty covers the detailed specs of most of the 7400 series TTL logic devices. No experimenter working with TTL has a complete library without **The TTL Data Book for Design Engineers.** Order yours today, only $3.95 postpaid.

● **The Supplement to The TTL Data Book for Design Engineers,** by Texas Instruments Incorporated. What happens when you can't find a 7400 series device listed in **The Data Book for Design Engineers?** Before you start screaming and tearing your hair out in frustration, turn to the Supplement. The Supplement has 400 pages of additional information including a comprehensive index to both TTL Data Book volumes. To be complete (and keep your hair in place and vocal cords intact) you'd best order the supplement at $1.95 to accompany the main volume.

● **The Linear and Interface Circuits Data Book for Design Engineers,** by Texas Instruments Incorporated. When you run across one of those weird numbers like 75365 the immediate frustration problem occurs again. What kind of gate could that be? We won't tell in this ad, but you can find out by reading the specifications in **The Linear and Interface Circuits Data Book for Design Engineers.** You can interface your brain to the 72xxx (linear) and 75xxx (interface) series of functions by ordering your copy of this 688 page manual at only $3.95 postpaid.

● **The Semiconductor Memory Data Book for Design Engineers,** by Texas Instruments, Incorporated. Don't forget the importance of memories to your systems. Refer to this 272 page manual to find out about the T.I. versions of many of the popular random access memories and read only memories. Order your personal copy today, only $2.95 postpaid.

● **The Transistor and Diode Data Book for Design Engineers,** by Texas Instruments, Incorporated. You'd expect a big fat data book and a wide line of diodes and transistors from a company which has been around from the start of semiconductors. Well, its available in the form of this 1248 page manual from T.I. which describes the characteristics of over 800 types of transistors and over 500 types of silicon diodes. This book covers the T.I. line of low power semiconductors (1 Watt or less). You won't find every type of transistor or diode in existence here, but you'll find most of the numbers used in switching and amplifying circuits. Order your copy today, only $4.95 postpaid.

● **The Power Semiconductor Handbook for Design Engineers** by Texas Instruments, Incorporated. To complement the low power transistor handbook, T.I. supplies this 800 page tome on high power transistors and related switching devices. Here is where you find data on the brute force monsters which are used to control many Watts electronically. Fill out your library with this book, available for only $3.95 postpaid.

● **Understanding Solid State Electronics** by Texas Instruments, Incorporated. This is an excellent tutorial introduction to the subject of transistor and diode circuitry. The book was created for the reader who wants or needs to understand electronics, but can't devote years to the study. This 242 page softbound book is a must addition to the beginner's library at only $2.95.

● **The Optoelectronics Data Book for Design Engineers** by Texas Instruments, Incorporated. This 366 page book is a compendium of information on T.I. phototransistors, LEDs and related devices. Order yours at $2.95 postpaid.

Buyers of these books should be cautioned: heavy reading will be required. These books are so filled with information that they weigh in at a total of about 190 ounces (5387 grams). On the basis of sheer mass, these books have got to be the bargain of the century. Make sure that you use a structurally sound book shelf and above all avoid dropping one of these books on your foot. But the mass of these books doesn't affect the bargain: we pay postage on all orders shipped to addresses in the USA and Canada, so the prices you see are the prices you pay. (That's only $.005 per gram on the average.)

____ TTL Cookbook @ $8.95
____ TTL Data Book @ $3.95
____ Supplement to TTL Data Book @ $1.95
____ Linear and Interface Circuits @ $3.95
____ Semiconductor Memory Data @ $2.95
____ Transistor and Diode Data Book @ $4.95
____ Understanding Solid State Electronics @ $2.95
____ Optoelectronics Data Book @ $2.95
____ Power Semiconductor Handbook @ $3.95

Please add 75 cents for postage and handling. Please allow six weeks for delivery.

Send to: Name ____________

Address ____________

City ________ State ________ Zip ________

☐ Check enclosed

☐ Bill MC # ____________ Exp. Date ________

☐ Bill BA # ____________ Exp. Date ________

Signature ____________

BYTE PETERBOROUGH, NH 03458

AMP L'ANNY Says

If you've got Ants in your plants trying to dig up a real components bargain -- Try Tri-Tek

MC14412 UNIVERSAL MODEM CHIP

MC14412 contains a complete FSK modulator and de-modulator compatible with foreign and USA communications. (0-600 BPS)

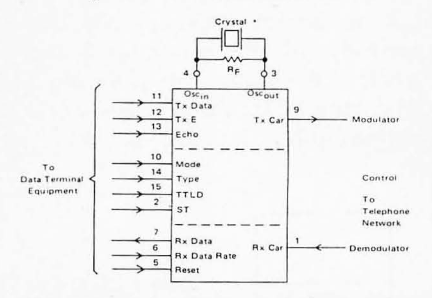
FEATURES:

- .On chip crystal oscillator
- .Echo suppressor disable tone generator
- .Originate and answer modes
- .Simplex, half-duplex, and full duplex operation
- .On chip sine wave
- .Modem self test mode
- .Selectable data rates: 0-200, 0-300, 0-600
- .Single supply
  VDD=4.75 to 15VDC - FL suffix
  VDD=4.75 to 6 VDC - VL suffix

TYPICAL APPLICATIONS:

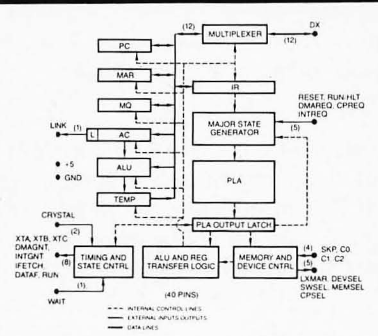
- .Stand alone - low speed modems
- .Built - in low speed modems
- .Remote terminals, accoustic couplers

MC14412FL.................................. $28.99
MC14412VL.................................. $21.74
6 pages of data......................... .60
Crystal for the above......$4.95



MC14411 BIT RATE GENERATOR.

Single chip for generating selectable frequencies for equipment in data communications such as TTY, printers, CRT s or microprocessors. Generates 14 different standard bit rates which are multiplied under external control to 1X, 8X, 16X or 64X initial value. Operates from single +5 volt supply. MC14411......................... $11.98
4 pages of data.................................. .40
Crystal for the above........................... $4.95



IM6100 CPU. Intersils' 12 bit CMOS CPU chip is the microprocessor which recognizes the famous PDP8/E instruction set. Single power supply, 4-7V@ 400uA. $52.50
Full data packet.................................$4.00

TELETYPE CODE CONVERSION CHIP

MM5220BL converts 5 level Baudot into 8 level ASCII. Use this chip to make your old TTY talk to your new computer.
MM5220BL.................................. $18.00
Specs for the above...................... .30

MOS TIME BASE KIT.

Only 1" X 1.5". Input 5 to 15 VDC, output is 60HZ square wave for portable or mobile clocks. PC board is drilled! MTBK-60HZ...................... $5.88



78H05 Voltage regulator. Fairchild 5V, 5A, TO-3 regulator. Take care of those heavy current requirments without separate regulator/pass transistor combinations. Use it with the same ease of instalation as the 309K(same pin arrangement.)..........with specs............... $11.25



LM317 Voltage Regulator. 1.5A, 3 terminal adjustable regulator in TO-3 case. Adjusts from +1.2V to +37V. Complete overload protection. .1% load regulation, .01%/V line regulation. No need to stock assorted regulators - just stock resistors......................$4.99
Specs for the above.................. .70

DATA BOOKS BY NATIONAL SEMICONDUCTOR

DIGITAL. Covers TTL, DTL, Tri-State, etc. $3.95
LINEAR. Covers amplifiers, pre-amps, op-amps, .. $3.95
LINEAR APPLICATIONS. Dozens of application notes and technical briefs covering the use of op-amps, regulators, phase locked loops and audio amps................ $3.25
CMOS Gates, Flip Flops, registers, functional blocks $3
VOLTAGE REGULATORS. A must for anyone making a power supply. Complete theory including transformers, filters, heat sinks, regulators, etc................. $3.00
MEMORY. Information on MOS and Bipolar memories: RAMS, ROMS, PROMS and decoders/encoders.. ... $3.95
INTERFACE. Covers peripheral drivers, level translators, line driver/receivers, memory and clock drivers, sense amps display driver and opto-couplers.................. $3.95
(Outside U.S., add postage for 1.5lbs)

DATA BOOKS FROM FAIRCHILD.

uA Linear. 776 pages of data and applications for Fairchild linear ICs. Great value...................$4.25
MOS/CMOS/nMOS/pMOS/CCD. Data and applications on MOS and charge coupled devices including preliminary data on new and future offerings. Want to know about 16K charge coupled line addressable memories?.........$3.95

RCA

GOLD CHIP

Linear Integrated Circuits

Brand new process by RCA in which the aluminum metalization has been replaced by gold. The chip is then hermetically sealed. What this means to you is unprecedented reliability and uniformity. Plastic parts that meet mil specs!!
Tri-Tek is proud to be the first to bring this new level of performance to you at SURPLUS PRICES. Why buy regrades??

CA301A..Improved,general purpose op-amp,8 pin dip..59¢
CA307...Super 741 op-amp. 8 pin dip...............52¢
CA324...Compensated quad op-amp, 14 pin dip.....$1.80
CA339A.. Low offset quad comparator. 14 pin dip...$1.59
CA741C..Famous general purpose op-amp, 8 pin dip.. 45¢
CA747C..General purpose dual op-amp, 14 pin dip... 82¢
CA748C..Externally compensated 741, 8 pin dip.......49¢
CA1458..General purpose dual op-amp. 8 pin dip.....69¢
CA3401..Quad single supply (5-18V) op amp. 14 pin..89¢

Another super buy from RCA. CA555 timer. 8 pin dip. 59¢

INTEL Data Catalog. Contains latest information on all the famous INTEL micro-processor and memories............$4.00

IMPROVED Performance version of the famous 8080, 8 bit micro processor. 8080A............................$34.95

SN75491....digit driver............................ 59¢
SN75492....digit driver............................ 59¢

UART AY5-1013A An ideal device for driving computer peripherals such as teletypes & video terminals. ...$9.45

PRECISION TO-92 plastic regulator. LM340AZ, available in +5 or +15 Volts. Improvement over the 78L05, same size.100mA.
LM340AZ-5 or LM340AZ-15.........................$1.10

NSL4944 LED. Current regulated, universal diffused-lens red LED lamp. A GaAsP solid-state high intensity LED encapsulated in a plastic package containing a current regulating IC that provides constant intensity over a wide voltage range. 2 to 18V, AC or DC. Use for indicator lamps, optical coupling, battery charging circuits, logic probes, almost any place you need a lamp. Long life, wide angle. No series resistor needed. Typical 13mA forward current.
NSL4944.......with panel mounting clip........... 89¢

D-A CONVERTER BY ZELTEX

8 bit precision hybrid circuit for use in controllers, timers, volt meters, etc. Molded plastic package with P.C. pins. Super buy on this better than usual subsystem. ZELTEX model ZD430. DAC-430.....................$4.95

NEW BOOK!!! "An Introduction to Microcomputers"
This is the book which Fairchild Semiconductor Company called ".....the best darned introduction to the industry to date." Covers everything from basic concepts to a review of real microcomputers. IMC-001.............$8.00



tri-tek, inc.
6522 north 43rd avenue,
glendale, arizona 85301
phone 602 - 931-6949

We pay shipping on all orders over $10 US, $15 foreign in US funds. Orders under $10, please add $1 handling. Please add insurance. Master Charge and Bank America cards welcome, ($20 minimum) Telephone orders may be placed 11AM to 5PM daily, Mon thru Fri. Call 602-931-4528. Check reader service card or send stamp for our latest flyers packed with new and surplus electronic components.

Software Bug of the Month I

Professor Floyd's Bug

W Douglas Maurer
University Library Room 634
George Washington University
Washington DC 20052

This bug is often used by Prof Robert W Floyd, of the Computer Science Department at Stanford University, to illustrate his theory of the proof of correctness of programs.

We are given an array A, let us say given by the FORTRAN language statement DIMENSION A(100). The elements of A are assumed to be stored in ascending order by value. Thus if I < J, then A(I) ≤ A(J), for all I and J from 1 through 100. We are trying to find an element X in this table by the method of repeatedly dividing the table in half.

Figure 1 shows how the method is supposed to work. We have three indices, I, J, and K. The index J is supposed to be halfway between I and K. By testing X against A(J), we can see whether X is between I and J or between J and K. If X is less than A(J), then X is between I and J; otherwise it is between J and K.

Initially, we set I = 1 and K = 100, so we are searching the whole table. At each stage, we divide the table in half, and set the new I and K to be the beginning and the end, respectively, of the new table (either the first half or the second half of the old table). When the table size gets down to 1, the algorithm stops, since we can now test a single element only. The FORTRAN expression of this program is as follows, assuming that, if X is in the table, we go to statement number 4, and otherwise we continue with the next statement in sequence:

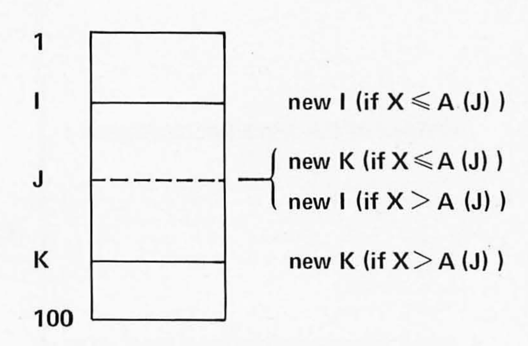


Figure 1: A sketch of the array A in which a particular value, X, is being sought.

```
      I = 1
      K = 100
1     J = (I + K) /2
      IF (X.GT.A(J) ) GO TO 2
      K = J
      GO TO 3
2     I = J
3     IF (I .NE. K) GO TO 1
      IF (X.EQ.A(I) ) GO TO 4
```

When this program is tried out, it works intermittently. Sometimes it finds the quantity X in the table, and sometimes it goes into an endless loop, even when X is in the table.

Can you find the bug?

SOLUTION IN NEXT MONTH'S BYTE



Here lies documentation of known bugs detected in previous editions of BYTE . . .

Apologies to author Roger Frank: The caption to listing 3, page 72, BYTE May 1976, should read

"A successive approximation conversion, specified as a symbolic assembly language program for the Motorola 6800 processor. Note that for fast processors or slow operational amplifiers (such as the 741), a delay loop should be inserted between lines 4 and 5 of this program to allow the output to settle."

Mr Frank had independently created his routine several months before publication of Motorola's application note on conversion techniques. . . . CH

My Dear Aunt Sally's Migraine

The example of table 2 in "My Dear Aunt Sally" (page 24, February 1976 BYTE) contains an error in the application of precedence rules. The last six lines of the table should read:

Input	Stack	Output
*	+↑(/*	
G	+↑(/*	G
)	+↑	*
end of string		/
		↑
		+

This error was detected by Roger Fritz of Davenport IA among others.

In J Bradley Flippin's "The SR-52: Another World's Smallest," April 1976, page 38: The expression given near the top of the second column has a single character error which completely changes the value computed. The error is due to a translation of a division sign into an addition sign. The correct expression is:

6 x (9÷ (6 x (12÷ (3 x (8 x (2 x (6÷ (6 x (6 + 2))))))))

This error was detected by Abijah Reed of Polaroid Corporation, Cambridge MA.■

S. D. SALES CO. P. O. BOX 28810 DALLAS, TEXAS 75228

ALARM CLOCK KIT SIX DIGIT LED

Thousands of hobbyists have bought and built our original clock kit and were completely satisfied. But we have received many requests for an alarm clock kit with the same value and quality that you have come to expect from S.D. So, here it is!

THE KIT INCLUDES:

- 1 Mostek 50252 Alarm Clock Chip
- 6 Hewlett Packard .30 in. common cathode readouts.
- 15 NPN Driver Transistors
- 1 Etched and Drilled P.C. Board set
- 1 Step Down Transformer
- 2 Switches for time set
- 2 Slide Switches for alarm set and enable
- 1 Filter Cap
- 4 IN4002 Rectifiers
- 1 IN914 Diode
- 1 .01 Disc Cap
- 15 Resistors
- 1 Speaker for alarm
- 1 LED lamp for PM indicator.

$16.50
(COMPLETE KIT)

Why pay MORE MONEY for our competitor's clock that has LESS DIGITS that are SMALLER in size?

Please take note that we use only first run parts in our kits and include ALL the necessary parts. Not like some of our competitors who use retested readouts and chips or who may not even include switches in their kits.

60 Hz. Crystal Time Base

FOR DIGITAL CLOCKS
S. D. SALES EXCLUSIVE!

$5.95

The kit you have been waiting for is here **NOW**, and at an unbelievable price! Thanks to S.D. Sales you can turn that digital clock of yours into a superbly accurate, DC operated, time piece.

KIT FEATURES:

A. 60 Hz output with accuracy comparable to a digital watch.
B. Directly interfaces with all MOS clock chips.
C. Super low power consumption (1.5 Ma typ.)
D. Uses latest MOS 17 stage divider IC.
E. Eliminates forever the problem of AC line glitches.
F. Perfect for cars, boats, campers, or even for portable clocks at ham field days.
G. Small size, can be used in existing enclosures.

BUY TWO FOR $10.00

Kit includes crystal, divider IC, P.C. Board plus all other necessary parts and specs.

ORDERS OVER $15 CHOOSE $1 FREE MERCHANDISE

2102 1K RAM's - 8 FOR $12.95

New units ——— We bought a load on a super deal, hence this fantastic price.
Units tested for 500NS Speed.

MOTOROLA RTL IC'S

Brand new, factory prime. Hard to find, but still used in a variety of projects. (See the RTL Cookbook by Howard W. Sams.)

MC724P-59c	MC780P-89c	MC791P-69c
MC725P-59c	MC785P-49c	MC792P-59c
MC764P-49c	MC787P-89c	MC799P-59c
MC767P-69c	MC788P-49c	MC9704P-89c
MC771P-49c	MC789P-59c	MC9709P-69c
MC775P-89c	MC790P-89c	MC9760P-69c

MV-50 TYPE LED's
by LITRONIX
10 for $1
Factory Prime!

3 DIGIT LED ARRAY — 75c
by LITRONIX
DL33MMB. 3 MAN-3 Size Readouts in one package. These are factory prime, not retested rejects as sold by others.
compare this price! 75c 3 for $2.

SALE ON CUT LEAD SEMICONDUCTORS

Leads were cut for PCB insertion. Still very useable.

1N914/1N4148 .... 100/$2
1N4002 1 Amp 100 PIV .... 40/$1
1N4745A 16V 1W Zener .... 20/$1
EN2222 NPN Transistor .... 25/$1
EN2907 PNP Transistor .... 25/$1
2N3904 NPN Driver Xstr. .... 25/$1
2N3392 GE Pre-amp Xstr. .... 25!$1
C103Y SCR. 800MA. 60V. .... 10/$1

ALL NEW. UNUSED. SOME ARE HOUSE #

SLIDE SWITCH ASSORTMENT

Our best seller. Includes miniature and standard sizes, single and multi-position units. All new, first quality, name brand switches. Try one package and you'll reorder more. *Special — 12 for $1* (Assortment)



DISC CAP ASSORTMENT

PC leads. At least 10 different values. Includes .001, .01, .05, plus other standard values.
60 FOR $1



UPRIGHT ELECTROLYTIC CAPS

47 mfd 35 V-10/$1 68 mfd 25V-8/$1
Brand new by Sprague. PC leads.

RESISTOR ASSORTMENT

1/4 W 5% and 10%. PC leads.
A good mix of values. 200/$2





1000 MFD FILTER CAPS

Rated 35 WVDC. Upright style with P.C. leads. Most popular value for hobbyists. Compare at up to $1.19 each from franchise type electronic parts stores. *S.D. Special 4 for $1*

FAIRCHILD BIG LED READOUTS

A big .50 inch easy to read character. Now available in either common anode or common cathode. Take your pick. Super low current drain, only 5 MA per segment typical.

FND - 510 Common Anode
FND - 503 Common Cathode

YOUR CHOICE
$1.50 ea. 6 for $7.50

DUAL 741C (5558) OP AMPS

Mini dip. New house numbered units by RAYTHEON.
4 FOR $1

FET'S BY TEXAS INSTRUMENTS — SPECIAL 5 for $1

#TIS-75 but with an internal house number. TO-92 plastic case. N. Channel, Junction type FET.

We do not sell junk. Money back guarantee on every item. No C.O.D. Texas Res. add 5% tax. Postage rates went up 30%! Please add 5% of your total order to help cover shipping.

ORDERS UNDER $10 ADD 75c HANDLING.

S. D. SALES CO.
P.O. BOX 28810
DALLAS, TEXAS 75228



Photo 1.

Systems of Note

Here is a prototype for a new BYTE feature which will appear regularly when material becomes available from our readers. The purpose of **Systems of Note** *is to document what kinds of systems are being developed and utilized by our readers, and how they are viewing the computer systems field as reflected in their choices of hardware and software components. As an example to illustrate the kind of short system description involved, I have written down a summary of my personal "BYTE #1" system as it stands March 17 1976.*

Each reader whose system description is published in this feature will receive an honorarium of $25 as BYTE's contribution to help further the state of the art. . . . **CH**

The BYTE #1 personal computing system is an M6800 processor built up into a general purpose computing system. The hardware of the processor, memory and peripherals is assembled with wire wrap prototyping cards available from Celdat Design Associates (Box 752, Amherst NH 03031). Its present (and projected) state can be seen by the following hardware map:

0000-00FF Page 0 memory; addresses 0000-000B reserved for soft interrupt vectors, stack starts at FF.

0100-013F Bootstrap ROM area. Presently has "soft" ROM made of 2102s and toggle switches which simulates two 8223s.

0140 Keyboard Scanner (ASCII Upper/Lower Case, home brew).

0142 UART Control: TTY (110 baud) or tape (faster) rates; master reset, etc.

0143 UART data.

0144 8 bit output latch (now connected to programmable counter as a kluge musical instrument).

014C-014F 32 bits of panel lamp latches on ASCII keyboard assembly.

*0150-0153 PIA 0 (16 bits of IO port).

*0154-0157 PIA 1 (16 bits of IO port).

*0158-0159 ACIA (General TTL and RS-232 interface for surplus ITT ASCIScope).

*015A ACIA data rate divide ratio transfer from PIA port.

1000-1FFF 4 K x 8 static 9102 memory.

*2000-3FFF 8 K x 8 static AM9140ADC memory (16 chips).

*4000-EFFF Expansion area, 4 K increments expected.

FFF0-FFFF ROM interrupt hardwired vectors, 8223 ROM. Reset vectors to location 0101 (low order bit is ROM burning mistake). All others vector to

SOLID STATE SALES



Features Value

SPECIAL PURPOSE TRANSISTORS

INTEGRATED CIRCUITS

HARDWARE
+
P.C. SOCKETS

SILICON POWER RECTIFIERS

ZENERS

TRIACS

SCRS

LIGHT EMITTING DIODES

SPECIAL PURCHASE ITEMS

WE HAVE AVAILABLE

USED IBM SELECTRIC 731 INPUT/OUTPUT TYPEWRITERS,

WITH MANUALS FOR $350.00

ALSO AVAILABLE USED KLEINSCHMIDT M-311 PRINTERS

THEY POINT AT THE RATE OF 120 CHARACTERS/SECOND.

$450.00

VIDEO CAMERA KIT

A UNIQUE ALL SOLID STATE CAMERA KIT
FEATURING A . . . 100 x 100 BIT
SELF SCANNING CHARGED
COUPLED DEVICE

INCLUDES THE FOLLOWING UNIQUE FEATURES . . . FOUND IN FAR MORE EXPENSIVE CAMERAS IF AVAILABLE

- LOW VOLTAGE SUPPLY (OR BATTERIES) + 5 AND + 15 VOLTS
- SENSITIVE TO INFRA RED AS WELL AS VISIBLE LIGHT
- MAY BE USED FOR IR SURVEILLANCE WITH AN IR LIGHT SOURCE
- EXCELLENT FOR STANDARD SURVEILLANCE WORK BECAUSE OF ITS LIGHTWEIGHT AND SMALL SIZE
- ALL COMPONENTS MOUNTED ON TWO PARALLEL 3"x5" SINGLE SIDED BOARDS TOTAL WEIGHT UNDER 2 LBS.
- MAY BE WIRED BY PERSON WITH SOME TECHNICAL EXPERIENCE IN 4-6 HRS.



SUPER UNBELIEVABLE

ONLY $225.00

MAY BE USED WITH AMATEUR RADIO FOR VIDEO

USED FOR CHARACTER RECOGNITION FOR COMPUTERS WITH EXTERNAL CIRCUITS

MAY BE USED IN A VACUUM, UNDER WATER, HIGH ALTITUDE, AND IN MAGNETIC ENVIRONMENT BECAUSE THERE IS NO HIGH VOLTAGE OR MAGNETIC DEFLECTION

WE SUPPLY ALL SEMICONDUCTORS, BOARDS, DATA SHEETS, DIAGRAMS, RESISTORS AND CAPACITORS

SORRY, WE DO NOT SUPPLY THE CASE, BATTERIES (OR SUPPLY) THE LENS (NOT SUPPLIED) DEPENDS UPON THE USE

ADD $2.00 POSTAGE AND HANDLING, PLUS $6.00 FOR ALL IC SOCKETS (Optional)

F8 MICROPROCESSOR KIT

WE'VE GOT THE F8 MICROPROCESSOR KIT, ONE OF THE MOST ADVANCED MCU SYSTEMS ON THE MARKET TODAY FOR ONLY **$159.00**

This three chip microprocessor system has the following advantages:
1) Driven by a +5 and +12 volt power supply.
2) Two I/O ports on the CPU chip, and ROM, making 32 bidirectional lines.
3) 64 bytes of fast RAM scratchpad built into the CPU chip.
4) A built in clock generator and power-on reset built into the CPU chip.
5) A programmable internal timer built into the ROM chip.
6) 60% of the instructions are 1 byte.
7) TTL I/O compatibility.
8) Consumes less than 300mw of power per chip.
9) A local interrupt with automatic adress vector.
10) Expandable to 64K bytes (2102-1's) of memory.
11) 20 mil loop and RS – 232 included.

The F8 Kit has enough parts and instructions to demonstrate microprocessor programs up to 1K byte, and to debug those programs.

We supply:
- 1 - 3850 CPU
- 1 - 3851 A FAIR-BUG programmed storage unit, provides the programmer with all its I/O subroutine, and allows the programmer to display or alter memory, and register contents via a teletype terminal.
- 1 - 3853 Static memory interface
- 8 - 2102

Plus CMOS gates and buffers, PC card, instruction manuals, programming guide, and time sharing guide.

4K MEMORY KIT $159.00

Memory Board: Our unique memory card features:
1) 8 bit bidirectional port, outputs buffered.
2) On board decoding for any four of 64 pages.
3) Address buffered.
4) 4K bytes of 2102-1 static RAM's.
5) No on board regulation to cause heat problems.

The memory card, like our other computer cards, has a 44 pin gold plated edge (.156 spacing). This feature makes the system completely stackable to save space, service problems, and hard wiring. It will also allow plug-in capability for our other cards when they become available.

EXPANDER BOARD

This board will expand the memory capability to 16 kilobytes with full buffering. Individual power terminals for each memory card are available. This system can be expanded to the full 64 kilobytes of memory by plugging other expander cards into this one with the 44 pin connector options.

To the best of our knowledge, ours is the only kit on the market designed for easy expansion to full memory capability.

EXPANDER KIT WITH ONE EDGE CONNECTOR $59.00 Extra 44 Pin Connectors

INTEL 8080 CPU $29.50
8008 8 BIT MICRO PROCESSING CHIP (with Data Book) $19.00
2102-1 1024 BT RAM . . . . $ 2.60
5202A UV PROM $12.50
MM5203 UV PROM. $12.50
1702A UV PROM $12.50
5204-4K PROM $24.95
MINIATURE MULTI-TURN TRIM POTS 100, 500, 5K, 10K, 25K, 50K, 100K, 200K $.75 each . . . . 3/$2.00
MULTI-TURN TRIM POTS Similar to Bourns 3010 style 3/16"x5/8"x1-1/4"; 50, 100, 1K, 10K, 50K ohms $1.50 ea. . . . . 3/$4.00
LIGHT ACTIVATED SCR's TO-18, 200V 1A. . . . . $ 1.75

PRINTED CIRCUIT BOARD

4-1/2"x6-1/2" SINGLE SIDED EPOXY BOARD 1/16" thick, unetched
$.50 ea. . . . . 5/$2.20
VECTOR BOARD 1" SPACING 4.5"x6.5" SHEET $1.50

4 WATT IR LASER DIODE $7.95

2N 5460 P FET . . . . $ .45
2N 5457 N FET . . . . $ .45
2N 4891 UJT. . . . . $ .45
TIS 43 UJT . . . . $ .35
ER 900 TRIGGER DIODES. . . . . 4/$1.00
2N 6028 PROG. UJT . . . . $ .65

VERIPAX PC BOARD

This board is a 1/16" single sided paper epoxy board, 4½"x6½" DRILLED and ETCHED which will hold up to 21 single 14 pin IC's or 8, 16, or LSI DIP IC's with busses for power supply connector. $5.25

MV 5691 YELLOW-GREEN BIPOLAR LED $1.25
MT-2 PHOTO TRANS . . . . $ .60
RED, YELLOW, GREEN OR AMBER LARGE LED's . . . ea. $ .20
14 PIN DIP SOCKETS . . . . $ .35
16 PIN DIP SOCKETS . . . . $ .35
MOLEX PINS . . . . 100/$1.00 1000/$8.00
8 PIN MINI DIP SOCKETS . . . . $ .30
10 PIN TO-5 TEFLON PC SOCKETS . $ .60
10 WATT ZENERS 3, 9, 4, 7, 18 OR 22V . . . . ea. $ .60
1 WATT ZENERS 4, 7, 5, 6, 10, 12, 15, 18 OR 22V . . . . ea. $ .25

Silicon Power Rectifiers

PRV	1A	3A	12A	50A	125A
100	.06	.14	.30	.80	
200	.07	.20	.35	1.15	4.25
400	.09	.25	.50	1.40	6.50
600	.11	.30	.70	1.80	8.50
800	.15	.35	.90	2.30	10.50
1000	.20	.45	1.10	2.75	12.50

REGULATED MODULAR POWER SUPPLIES

+ – 15 VDC AT 100ma 115VAC INPUT $27.95
5VDC AT 1A. 115VAC INPUT . . . $24.95
12 VDC AT .54 . . . . $24.95
IN 4148 (IN914) . . . . 15/$1.00
2518-HEX 32 BIT SR. $6.00
SILICON SOLAR CELLS 2¼" diameter .5V at 500 ma. . . . . $5.00 ea., 6/$27.50
7 POLE 1 THROW TO-5 MINIATURE ROTARY SWITCH . . . . $ .95

TRANSISTOR SPECIALS

2N3585 NPN Si TO-66 . . . . $ .95
2N3772 NPN Si TO-3 . . . . $ 1.80
2N4901 PNP Si TO-3 . . . . $ .85
2N5086 PNP Si TO-92 . . . . 4/$ 1.00
2N4898 PNP TO-66 . . . . $ .60
2N404 PNP GE TO-5 . . . . . . 5/$ 1.00
2N3919 NPN Si TO-3 RF . . . . . . $ 1.50
MPSA 12 NPN Si TO-92 . . . . . 3/$ 1.00
2N3767 NPN Si TO-66 . . . . $ .70
2N2222 NPN Si TO-18 . . . . 5/$ 1.00
2N3055 NPN Si TO-3 . . . . $ .80
2N3904 NPN Si TO-92 . . . . 5/$ 1.00
2N3906 PNP Si TO-92 . . . . 5/$ 1.00
2N5296 NPN Si TO-220 . . . . $ .50
2N6109 PNP Si TO-220. . . . . $ .55
2N3866 NPN Si TO-5 . . . . $ .75
2N3638 NPN Si TO-5 . . . . 5/$ 1.00
2N6517 NPN TO-92 Si . . . . 3/$ 1.00

TANTULUM CAPACITORS

.22UF 35V	5/$1.00	6.8UF 35V	3/$1.00
.47UF 35V	5/$1.00	33UF 25V	$ .40
.68UF 35V	5/$1.00	30UF 6V	5/$1.00
1UF 35V	5/$1.00	150UF 20V	$ .50
4.7UF 35V	4/$1.00		

CT7001 ALARM CLOCK CHIP $5.75
FPA 711-7 LEVEL Diode Array Optical Tape Readers $5.95

NATIONAL MOS DEVICES

MM1402–3.20	MM5057–4.00
MM1403–3.20	MM5058–4.95
MM1404–2.50	MM5060–4.95
MM5013–7.75	MM5061–4.30
MM5016–3.50	MM5555–6.25
MM5017–4.75	MM5556–6.25
MM5055–4.00	MM5210–1.95
MM5056–4.00	MM5260–2.95

TTL IC SERIES

74L00– .30	7483– .90
7400– .18	7485–1.05
7401– .18	7486– .45
7402– .18	7489–1.80
7403– .18	7490– .60
7404– .22	7491– .79
7405– .22	7492– .60
7406– .35	7493– .60
7407– .33	7494– .80
7408– .22	7495– .85
7409– .25	7496– .79
7410– .18	74107– .34
7411– .25	74121– .50
7412– .30	74123– .85
7413– .60	74125– .70
7414–1.45	74126– .90
7416– .33	74150–1.00
7417– .33	74151– .90
7420– .18	74153– .79
7425– .35	74154–1.40
7426– .35	74155–1.40
7427– .35	74157– .75
7430– .18	74161–1.20
7432– .24	74164–1.50
7437– .35	74165–1.50
7438– .35	74173–1.35
7440– .18	74175– .95
7441– .95	74177–1.00
7442– .70	74180–1.05
7445– .85	74181–2.30
7446–1.00	74191–1.50
7447– .87	74192–1.10
7448–1.00	74193–1.25
7472– .35	74194–1.25
7473– .40	74195– .74
7474– .40	74196–1.20
7475– .60	75324–1.75
7476– .40	75491– .80
7480– .48	75492– .80

ALCO MINIATURE TOGGLE SWITCHES
MTA 106 SPDT $1.20
MTA 206 DPDT $1.70

C/MOS (DIODE CLAMPED)

74C02– .30	4016– .60	4035–1.75
74C01– .30	4017–1.30	4042–1.00
4001– .30	4018–1.40	4046–2.75
4002– .30	4019– .60	4047–3.00
4006–1.60	4022–1.40	4049– .75
4007– .30	4023– .30	4050– .75
4009– .60	4024–1.25	4055–1.95
4010– .60	4025– .30	4066–1.10
4011– .30	4027– .75	4071– .30
4012– .30	4028–1.20	4077– .30
4013– .50	4029–1.45	4081– .30
4015–1.40	4030– .55	

LED READOUTS

FND 500-.5" C.C. $1.75
HP 7740-.3" C.C. $1.40
MAN-4-.25" C.C. $1.20
MAN-7-.3" C.A. $1.25
DL747-.6" C.A. $1.95
NS 33-3 dig. array $1.35

Full Wave Bridges

PRV	2A	6A	25A
200	.95	1.25	2.00
400	1.15	1.50	3.00
600	1.35	1.75	4.00

CD 110 LINEAR 256 XI BIT SELF SCANNING CHARGED COUPLED DEVISE $125.00

SANKEN AUDIO POWER AMPS

Si 1010 G 10 WATTS. . . . . $ 6.90
Si 1020 G 20 WATTS. $13.95
Si 1050 G 50 WATTS. $24.95

LINEAR CIRCUITS

LM 309K 5V 1A REGULATOR . . . $1.00
723 – 40 + 40VV REGULATOR . . . $ .54
301/748-Hi Per. Op. Amp. . . . . $ .35
320T 5, 12, 15, OR 24V NEGATIVE REG. $1.35
741A or 741C OP AMP. . . . . $ .35
709c OPER. AMP. . . . . $ .32
307 OP AMP . . . . $ .25
CA 3047 Hi Pef. Op. Amp. . . . . . . $ .95
340T 5, 6, 8, 12, 15, 18, 24V POS REG. TO-220 . . . . $1.20
101 OPER. AMP. HI PERFORM. . . . $ .75
LM 308 Oper. Amp., Low Power . . . $1.05
747 – DUAL 741 . . . . $ .65
556 – DUAL TIMER . . . . $ .95
537 – PRECISION OP. AMP. $2.80
540-70W POWER DRIVE $2.75
LM 3900 – QUAD OP. AMP . . . . . $ .49
LM 324 – QUAD 741 . . . . $1.50
560 – PHASE LOCK LOOP $2.50
561 – PHASE LOCK LOOP $2.50
565 – PHASE LOCK LOOP $1.50
566 FUNCTION GEN. $2.50
567 – TONE DECODER $2.00
LM 1310N FM STEREO DEMOD. . . $2.75
8038 IC VOLTAGE CONT. OSC. . . $3.90
LM 370 – AGC SQUELCH AMP. . . $1.15
555 – 2μs – 2 HR. TIMER. . . . . . . $ .53
553 QUAD TIMER $2.50
FCD 810 OPTO-ISOLATOR . . . . . $ .80
1458 DUAL OP AMP. . . . . $ .55
LM 380 – 2W AUDIO AMP. . . . . . $ .95
LM 377 – 2W Stereo Audio Amp. . . $2.50
LM 381 – STEREO PREAMP. $1.25
LM 382 – DUAL AUDIO PREAMP . $1.25
LM 311 – HI PER. COMPARATOR . $ .95
LM 319 – Dual Hi Speed Comp. . . . $1.25
LM 339 – QUAD COMPARATOR . $1.40

TRIACS / SCR'S

PRV	1A (TRIACS)	10A (TRIACS)	25A (TRIACS)	1.5A (SCR'S)	6A (SCR'S)	35A (SCR'S)
100	.40	.70	1.30	.40	.50	1.20
200	.70	1.10	1.75	.60	.70	1.60
400	1.10	1.60	2.60	1.00	1.20	2.20
600	1.70	2.30	3.60		1.50	3.00

Send 25¢ for our catalog featuring Transistors and Rectifiers
145 Hampshire St., Cambridge, Mass.

Terms: FOB Cambridge, Mass. Send Check or Money Order. Include Postage, Minimum Order $5.00, CPD'S $20.00



SOLID STATE SALES
P.O. BOX 74B
SOMERVILLE, MASS. 02143 TEL. (617) 547-4005

WE SHIP OVER 95% OF OUR ORDERS THE DAY WE RECEIVE THEM

JMP instructions in locations 0000-000B. In "run" mode, reset vectors to 0000 by switching ROM address bit.

(* indicates construction in progress.)

Peripherals at present include: surplus Teletype KSR 33, a modified BYTE standard tape interface (higher speed, higher density), the aforementioned programmable counter "music synthesizer" and keyboard. Peripherals for inclusion in the near future include:

- BYTE standard interface (from Harold Mauch).
- Single channel music peripheral.
- ASCISCOPE VDT as main control keyboard and console.

Software at present includes the IMP (Interactive Manipulator Program) monitor, the Kluge Harp Version 2 Music Interpreter (features subroutines, calls, special functions such as trill, chromatic run, tempo change), and an income tax accounting program which tabulates amounts by category (2 digit ASCII code).

The present IMP 6800 program takes about 700 bytes and uses the panel lamp displays to show address (16 bits), data at address (8 bits) and the last 8 bits of the current hexadecimal encoded keyboard entry. Functions include memory editing and address pointer manipulations, block write, block read, block to block compare, tape to block verification, block move, block zap, formatted object code dump (pays attention to instruction length), call program, etc. Version 2 will augment the present panel display with outputs to the ASCIScope peripheral, giving much greater detail of internal control variable values.

The most unusual peripheral of the system is an 8 by 4 feet (2.44 by 1.22 m) frame of wood suspended beneath the ceiling over the lab bench, with polyethylene plastic stapled to it. Purpose: prevent leaks due to ice buildup from soaking the processor.

Photo 1 shows the current state of the system, spread out on the top of an old conference table purchased for "peanuts" at a traditional New England style bargain source. Photos 2A and B illustrate construction details of the unique environmental protection unit which is expected to become obsolete in the near future when a house with sound roof is purchased to replace the present rented abode.■

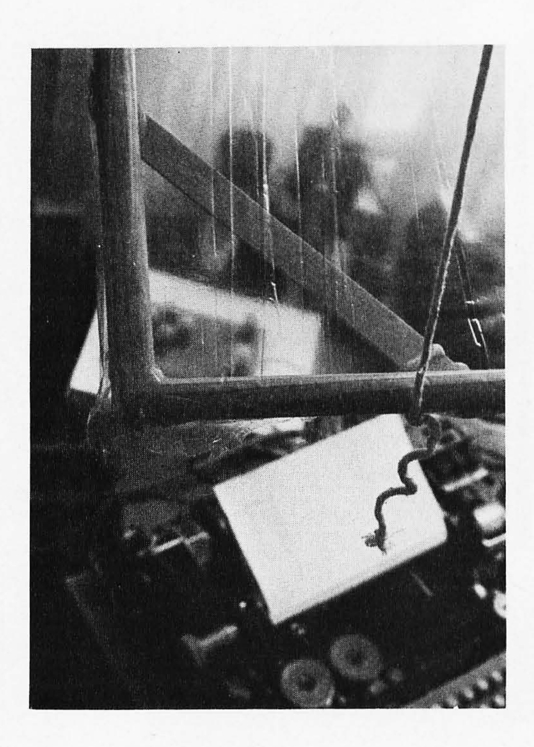


Photo 2A.

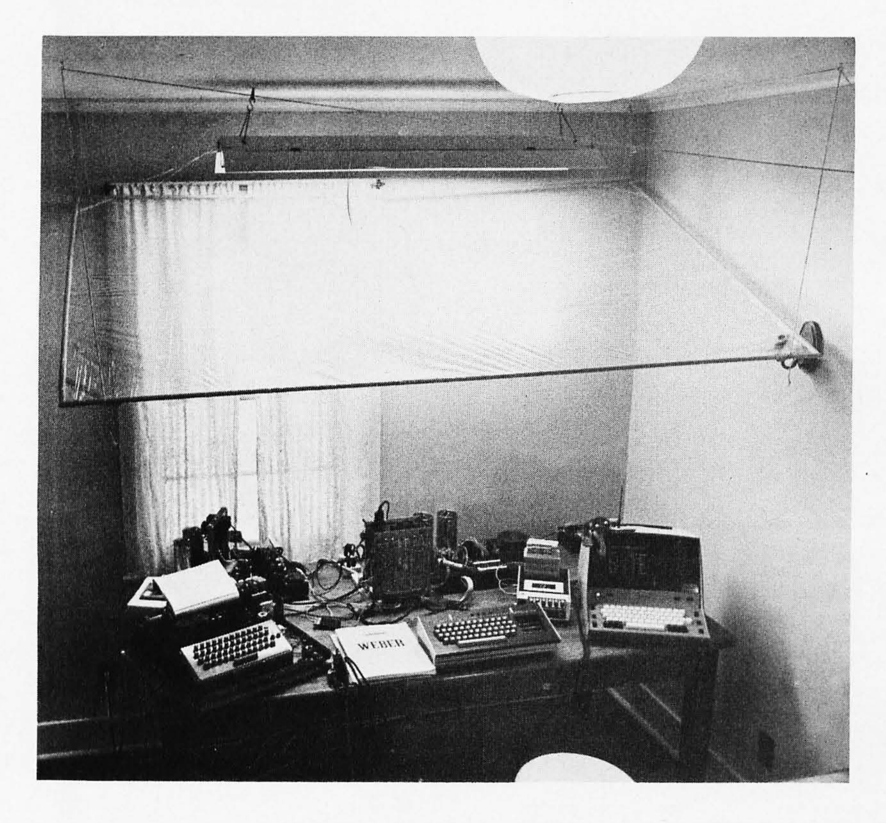
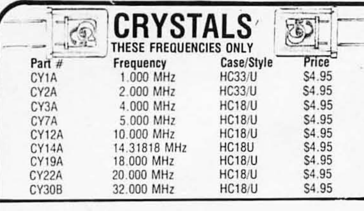


Photo 2B.



CRYSTALS

THESE FREQUENCIES ONLY

Part #	Frequency	Case/Style	Price
CY1A	1.000 MHz	HC33/U	$4.95
CY2A	2.000 MHz	HC33/U	$4.95
CY3A	4.000 MHz	HC18/U	$4.95
CY7A	5.000 MHz	HC18/U	$4.95
CY12A	10.000 MHz	HC18/U	$4.95
CY14A	14.31818 MHz	HC18U	$4.95
CY19A	18.000 MHz	HC18/U	$4.95
CY22A	20.000 MHz	HC18/U	$4.95
CY30B	32.000 MHz	HC18/U	$4.95

CLOCK CHIPS — CALCULATOR CHIPS

MM5309	6 Digit, BCD Outputs, Reset PIN	$5.95
MM5311	6 Digit, BCD Outputs, 12 or 24 Hour	4.95
MM5312	4 Digit, BCD Outputs, 1 PPS Output	4.95
MM5313	6 Digit, BCD Outputs, 1PPS Output	4.95
MM5314	6 Digit, 12 or 24 Hour, 50 or 60 Hz	4.95
MM5316	4 Digit, Alarm, 1PPS Output	6.95
MM5318	Video Clock Chip, For Use With MM5841	9.95
CT7001	6 Digit, Calander, Alarm, 12 or 24 Hour	6.95

CALCULATOR CHIPS

MM5725	6 Digit, Four Function, Less Decimal	$2.95
MM5738	8 Digit, 5 Function, +, =, ×, ÷, %	2.95
MM5739	8 Digit, 4 Function, Floating Decimal	2.95
CT5001	12 Digit, 4 Function	3.95
CT5005	12 Digit, 4 Function with Memory	5.95
CT5030	12 Digit, 4 Function and %	7.95

MISC. MOS

MM5320	TV Camera Sync. Generator	$19.95
MM5330	4½ Digit DVM Chip	9.95
MM5369	60 Hz Timebase Circuit From 3.58 MHz	5.00
MM5841	Video Generator For MM5318	18.00
MC1408-L7	7 Bit Digital to Analog Converter	9.95
MK5007	4 Decade Counter with Latches	10.95
LD110/LD111	3½ Digit DVM Chip Set	25.00
95H90	100 MHz ÷ 10 Counter For Prescalers	13.95

Continental Specialties

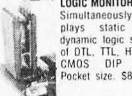
PROTO BOARDS

Build & test circuits as fast as you think!

PB100	10 IC cap breadboard kit, 4.5 x 6.0 x 1.35"	$19.95
PB101	10 14-DIP cap, 5-way post, 940 solderless tie points, 5.8 x 4.5"	$29.95
PB102	12 14-DIP cap., like PB101 with 1,240 tie points, 7.0 x 4.5"	$39.95
PB103	24 14-DIP cap., 4 5-way posts, 2,250 tie points, 6.0 x 9.0"	$59.95
PB104	32 14-DIP cap., 3060 solderless tie points, 8.0 x 9.76"	$79.95

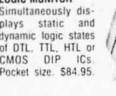
PROTO-CLIP

For power-on/hands-off signal tracing. Bring IC leads up from PC board surface for fast troubleshooting.
PC14 14-pin $4.50
PC16 16-pin 4.75



LOGIC MONITOR

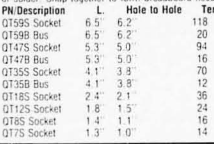
Simultaneously displays static and dynamic logic states of DTL, TTL, HTL or CMOS DIP ICs. Pocket size. $84.95.



SOCKETS & BUS STRIPS

Plug-in wire, test, modify or expand without patch cords or solder. Snap together to form breadboard needed.

PN/Description	L.	Hole to Hole	Term'ls	Price
QT59S Socket	6.5"	6.2"	118	$12.50
QT59B Bus	6.5"	6.2"	20	2.50
QT47S Socket	5.3"	5.0"	94	10.00
QT47B Bus	5.3"	5.0"	16	2.25
QT35S Socket	4.1"	3.8"	70	8.50
QT35B Bus	4.1"	3.8"	12	2.00
QT18S Socket	2.4"	2.1"	36	4.75
QT12S Socket	1.8"	1.5"	24	3.75
QT8S Socket	1.4"	1.1"	16	3.25
QT7S Socket	1.3"	1.0"	14	3.00





1/16 VECTOR BOARD

	0.1" Hole Spacing Part No.	P-Pattern L	W	Price 1-19	20-49
PHENOLIC	64P44 062XXXP	4.50	6.50	1.72	1.54
	169P44 02XXXP	4.50	17.00	3.69	3.32
EPOXY GLASS	64P44 062	4.50	6.50	2.07	1.86
	84P44 062	4.50	8.50	2.56	2.31
	169P44 062	4.50	17.00	5.04	4.53
	169P84 062	8.50	17.00	9.23	8.26
EPOXY GLASS COPPER CLAD	169P44 062C1	4.50	17.00	6.80	6.12

VECTOR TERMINALS

Series	Part No. Finish Gold	Tinned	Post Size	25 pcs Gold	Tin	50 pcs Gold	Tin
T44	T44-1	T44	.025" sq.	1.50 lot	1.00 lot	2.75 lot	1.75 lot
T46	T46-1	T46	.028" sq.	1.75 lot	1.25 lot	3.00 lot	2.25 lot
T49	T49A	T49	.025" sq.	1.75 lot	1.25 lot	3.10 lot	2.25 lot

VECTOR WIRING PENCIL

Vector Wiring Pencil P173 consists of a hand held featherweight (under one ounce) tool which is used to guide and wrap insulated wire, fed off a self-contained replaceable bobbin, onto component leads or terminals installed on pre-punched "P" Pattern Vectorbord. Connections between the wrapped wire and component leads, pads or terminals are made by soldering. Complete with 250 FT of red wire. **$9.50**

REPLACEMENT WIRE — BOBBINS FOR WIRING PENCIL

W36-3-A-Pkg. 3	250 ft. 36 AWG GREEN	$2.40
W36-3-B-Pkg. 3	250 ft. 36 AWG RED	$2.40
W36-3-C-Pkg. 3	250 ft. 36 AWG CLEAR	$2.40
W36-3-D-Pkg. 3	250 ft. 36 AWG BLUE	$2.40

100 PER STRIP

MOLEX PINS

Intended for use as an inexpensive substitute for IC sockets. Also perfect for use as board connectors and in subassemblies.

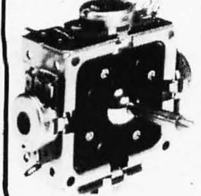
SPECIAL — 100/1.49 — 1000/12.00

64 KEY KEYBOARD



The ideal item for the real homebrew computer hobbyist. This keyboard features 64 unencoded SPST keys, unattached to any kind of P.C.B. A very solid molded plastic 13" x 4" base suits most applications.

$24.95 ea.



JOYSTICK

These joysticks feature four 100K potentiometers, that vary resistance proportional to the angle of the stick. Sturdy metal construction with plastics components only at the movable joint. Perfect for electronic games and instrumentation.

$9.95 ea.

MICROPROCESSOR COMPONENTS

8080 SUPPORT DEVICES

8080A $37.95

8212	8 BIT INPUT/OUTPUT PORT FOR 8080	$ 5.95
8224	CLOCK GENERATOR AND DRIVER FOR 8080	12.95
8228	SYSTEM CONTROLLER AND BUS DRIVER FOR 8080	12.95

8080 $24.95

	CPU'S	
8008	8 BIT CPU	$19.95
8080	Super 8008	24.95
8080A	Super 8008	37.95
	SR'S	
2504	1024 Dynamic	$ 9.00
2518	Hex 32 BIT	7.00
2519	Hex 40 BIT	4.00
2524	512 Dynamic	2.95
2525	1024 Dynamic	6.00
2527	Dual 256 BIT	3.95
2529	Dual 512 BIT	4.00
2532	Quad 80 BIT	3.95
2533	1024 Static	7.95
3341	Fifo	6.95
74LS670	16 x 4 Reg.	3.95
	UART'S	
AY-5-1013	20K Baud	$ 6.95
	ROM'S	
2513	Char. Gen.	$11.00
7488	Random Bits	3.50

		RAM'S	
3601	256 x 1	FAST	5.00
1101	256 x 1	Static	$ 2.25
1103	102 x 1	Dynamic	2.95
2101	256 x 1	Static	6.95
2102	1024 x 1	Static	1.95
2107	4096 x 1	Dynamic	19.95
2111	256 x 4	Static	7.95
7010	1024 x 1	MNOS	29.95
7489	16 x 4	Static	2.49
8101	256 x 4	Static	7.95
8111	256 x 4	Static	7.95
8599	16 x 4	Static	3.49
91L02	1024 x 1	Static	2.49
74200	256 x 1	Static	6.95
93410	256 x 1	Static	1.75
5262	2048 x 1	Dynamic	2.95
		PROMS	
1024	1024 x 1	HPROM	5.00
1702A	2048	Famos	$15.95
5203	2048	Famos	14.95
82S23	32 x 8	Open C.	5.00
82S123	32 x 8	Tristate	5.00

BUILD YOUR OWN JOLT MICROCOMPUTER IN JUST 3 HOURS OR LESS FOR $159.95

A COMPLETE MICROCOMPUTER IN A SINGLE CPU KIT INCLUDES: • An MOS Technology MCS6502 NMOS microporcessor • 512 bytes of program RAM, and 64 bytes of interrupt vector RAM • 1K bytes of mask programmed ROM containing DEMON, a powerful debug monitor • 26 programmable I/O lines • Internal RC clock, or crystal controlled clock with user supplied crystal • Serial I/O ports for use with a teleprinter current loop drive/receiver, or an EIA standard driver/receiver • Expandable address and data buses • Hardware interrupt • Control panel interface lines available on card connector • Complete assembly manuals and sample programs

JOLT ACCESSORY KITS

JOLT RAM CARD — Fully static 4,096 bytes of RAM with 1 microsecond access time and on-board decoding. **$199.95**

JOLT I/O Card (Peripheral Interface Adapter) — 2 PIA LSI chips, 32 I/O lines, four interrupt lines, on-board decoding and standard TIL drive. Fully programmable. **$95.50**

JOLT Power Supply — Operates at +5, +12 and −10 voltages. Supports JOLT CPU, 4K bytes of RAM and JOLT I/O card — or, CPU and 8 I/O cards. **$99.95**

JOLT +5V Booster Option — Fits onto JOLT Power Supply card. Supports CPU, 8K bytes RAM and 8 I/O CPU and cards. **$24.95**

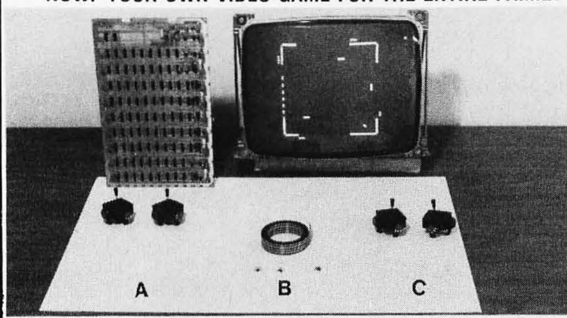
JOLT Universal Card — Sames size (4¼" x 7"), same form factor as other JOLT cards. Completely blank, drilled to accept 14, 16, 24 or 40 pin sockets. **24.95**

JOLT Accessory Bag — Contains enough hardware to connect one JOLT card to another, flat cable, connectors, card spacers, hardware, etc. **$39.95**

JOLT Resident Assembler — Fully symbolic, single pass resident assembler, all memonics compatible with timesharing assemblers. Delivered on four 1702A PROMs, ready for plugging into JOLT PROM card. **$149.95**

JOLT 1702A PROM Card — Sockets for 2,048 bytes of PROM memory. Place anywhere in memory with jumper selectable addresses. **$99.95**

NOW: YOUR OWN VIDEO GAME FOR THE ENTIRE FAMILY



This game comes pre-tested with two **PROFESSIONAL** Kraft joysticks. Joysticks allow 2 dimensional player control (rather than only one dimension, such as up and down.) If you require more than two players, order extra joysicks. All that's required is a 5V/2A power supply, a harness, and speaker. Comes with schematics, wiring information, and all necessary documentation, Game gives 1V composite video output, perfect for any TV monitor, Game designed so one, two, three, or four players can play at the same time. You can even play against the HOUSE. Score for each person is shown on TV set. These boards are production over runs of a well known video game manufacturer, and are **not** rejects, or in any way inferior to one presently being sold in games for over $1,000.00.

KIT A — $179.95 **PROFESSIONAL** game P.C. board, and 2 **PROFESSIONAL** joysticks. P.C. board size is 10½" x 17". This is the same **PROFESSIONAL** game as seen in commercial establishments. Dont't confuse it with the simple games sold in stores, or with analog kits.

ACCESSORY B — $3.95 Six feet of ribbon cable, three SPST switches (coin simulator, house/player, and start switch), for those of you who don't have extra wire or switches to build the harness.

ACCESSORY·C — $39.95 Two additional **PROFESSIONAL** Kraft joysticks, for third and fourth player.

DIGITAL CLOCK KIT — 3½ INCH DIGITS



This clock features big 3½" high digits for viewing in offices, auditoriums, etc. Each digit is formed by 31 bright 0.2" LED's. The clock operates from 117 VAC, has either 12 or 24 hr. operation. The 6 digit version is 27" x 3½" x 1½", and the 4 digit is 18" x 3½" x 1½". Kits come complete with all components, case and transformer.

Specify 12 Or 24 Hr. When Ordering

4 DIGIT KIT $49.95	4 DIGIT ASSEMBLED $59.95
6 DIGIT KIT $69.95	6 DIGIT ASSEMBLED $79.95

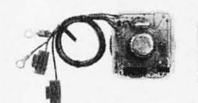
Satisfaction Guaranteed. $5.00 Min. Order. U.S. Funds.
California Residents — Add 6% Sales Tax — Data Sheets 25c each
Send a 13c Stamp (postage) for a FREE 1976 Catalog



1021 HOWARD ST., SAN CARLOS, CA. 94070
PHONE ORDERS — (415) 592-8097

AUTOTEL ™

WARNING LIGHTS ARE NOT ENOUGH





AN AUDIBLE ALARM INDICATING POTENTIAL ENGINE DAMAGE

AUTOTEL is an effecient (15 ma current standby) device by which every owner of an automobile, truck or vehicle equipped with indicator lights for temperature and oil pressure can be assured of a reliable warning **before** an impending failure.

AUTOTEL, by means of an audible signal 70 db pulsing) immediately forewarns the vehicle operator of a malfunction or failure, allowing time to correct and prevent major engine failures. It is programmed so there is no sound during normal operating conditions.

AUTOTEL features CMOS circuitry, packaged in a 2½" sq. x ⅞" case. The kit comes complete with all components, hardware and case to hook directly into your car's warning light system.

$14.95 Assembled $9.95 Per Kit

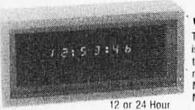
iMC 3½ DIGIT DVM KIT



This 0-2 VDC .05 per cent digital voltmeter features the Motorola 3½ digit DVM chip set. It has a .4" LED display and operates from a single +5V power supply. The unit is provided complete with an injection molded black plastic case complete with Bezel. An optional power supply is available which fits into the same case as the 0-2V DVM allowing 117 VAC operation.

A. 0-2V DVM with Case	**$49.95**
B. 5V Power Supply	**$14.95**

JE700 CLOCK



12 or 24 Hour

115 VAC

The JE700 is a low cost digital clock, but is a very high quality unit. The unit features a simulated walnut case with dimensions of 6" x 2½" x 1". It utilizes a MAN72 high brightness readout, and the MM5314 clock chip.

$19.95

Liquid Crystal Temperature Display

Six Digit Light Emitting Diode Display

DL 33 Display



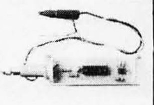
This clock makes an attractive addition to any desk. It has an extruded, black anodized aluminum case. It displays hours, minutes, and seconds with .11 inch displays, and comes complete with a liquid crystal thermometer. It operates off 115 VAC at 50 or 60 Hz. **$24.95**



This large digit clock (.6" hours & minutes, .3" seconds) features the MM5314 clock chip. It operates from 117 VAC, and will operate in either a 12 or 24 hour mode. The clock is complete with a walnut grain case, and has fast set, slow set, and hold time set features.

KIT - ALL COMPONENTS & CASE $34.95
WIRED & ASSEMBLED $39.95

JE803 PROBE



The Logic Probe is a unit which is for the most part indespensible in trouble shooting logic families TTL, DTL, RTL, CMOS. It derives the power it needs to operate directly off of the circuit under test, drawing a scant 10 mA max. It uses a MAN3 readout to indicate any of the following states by these symbols. (H) - 1 (LOW) - o (PULSE) - P. The Probe can detect high frequency pulses to 45 MHz. It can t be used at MOS levels or circuit damage will result.

$9.95 Per Kit
printed circuit board

ELECTRONIC TOOLS

QUICK CHARGE CORDLESS SOLDERING IRON

This versatile tool weighs only six ounces, and can be operated without power anywhere. It gives the performance of a 50 watt iron with a tip temperature of over 700°F. It will solder an average of 160 3 twist 22 guage wire joints on one charge. It has a charge time of 4 hours, and heats in 5 seconds. Complete with recharger.

$29.95 EACH

NIBBLING TOOL

This Nibbling Tool is perfect for cutting, trimming or notching sheet metal up to 18 guage. It operates like a punch and die and also works well on aluminum or plastic up to 1/16" thick. We feel that this tool can be a real time-saver for our customers when working on chassis, printed circuit boards and prototype motel parts.

Nibbling Tool	**$6.95**
Replacement Punch	**$3.75 Each**

DIAGONAL CUTTER

½" Semi-Flush Cut Fine Diagonal

4-in. long. Most popular cutter on assembly line cuts at extreme tip. Semi-flush cut. For fine wire work only. No 20 soft copper wire and smaller. Width across joint 13/32", thru joint 7/32", tip length 1/2".

Light Blue Handle
A97MS **$8.50 ea.**

SOLDER

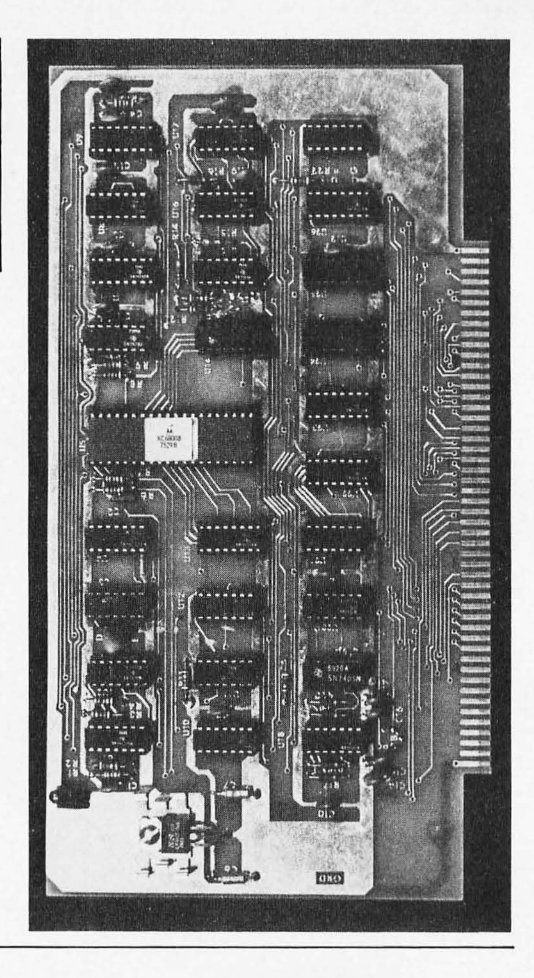


Part No.	Wire Dia.	Price/1 lb Spool
SN60	.031	6.85/Spool
SN60	.040	6.50/Spool

Solder Wick

STOCK	WIDTH	PRICE
40-1	.025	$2.00 ea
40-2	.050	2.00 ea
40-3	.075	2.00 ea

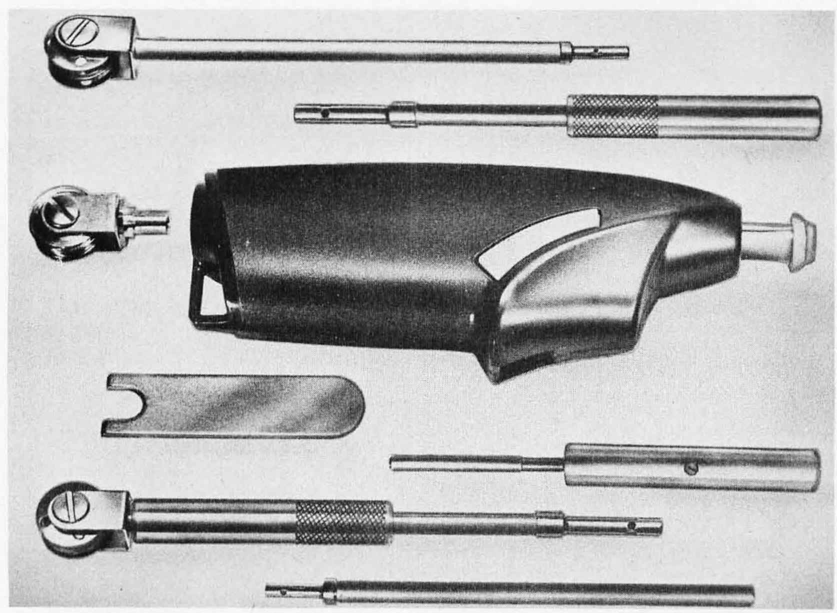
What's New?



An M6800 for an Altair 8800, Anyone?

Latest in the world of Altair compatible "peripherals" is this AM6800 board product announced by MRS, Hawthorne CA. The card plugs directly into any existing Altair 8800 main frame (also, by implication, the various Altair compatible main frames from alternate sources) and is designed to allow a "hand off" of control between the 8080 and 6800 CPUs, using a single instruction. The information BYTE received also states that there are no modifications required for use with an Altair 8800, and that the board will not interfere with the normal execution of the 8080. The board has been designed with attention to the speed of various memories available, so that it will operate with either fast or slow, dynamic or static memories available for the Altair. The 6800 CPU's status signals are brought out through jumpers to unused bus lines of the Altair backplane.

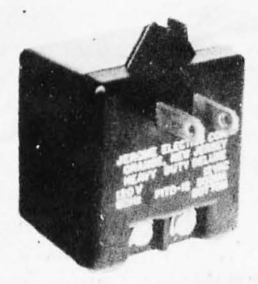
This product looks like an ideal one for individuals who want to benchmark both processors for an engineering design comparison, and for people engaged in professional consulting work who need to have both CPUs around to handle customer preferences. (When will MRS come out with the MOS Technology 6502 version?) MRS is located at PO Box 1220, Hawthorne CA 90250.■



Vector Electronic Company sent BYTE this photograph of a new line of inexpensive wire wrapping tools which include the P160-4R power driver in the center of this photograph, plus accessories which enable the user to hand or machine wrap wire from 22 to 30 gauge on wire wrap posts of several sizes. Priced at under $50 for the P160-4R hand held battery operated power driver, this family of tools provides one excellent low cost method of accomplishing wire wrap interconnection. For more information contact Vector Electronic Co Inc, 12460 Gladstone Av, Sylmar CA 91342, (213)365-9661.

Computer Take Over . . . continued

Jack Fostnaught, who owns a computer named RODNE (Robotic Omnipotent Device Not Entirely), had this story to tell of a computer's self serving mentality. It seems that a computer was installed in the Fourth National Bank Building in Wichita KS; its duties included, among other things, controlling the air conditioning of the entire building. One day two of the air conditioning units broke down necessitating a decision as to what part of the building could continue to be comfortably cool and what part would have to endure the heat. The computer thought things over briefly and then switched the remaining cooling power to its own room.



POWER SUPPLY MODULE

New, plug-in module. Plugs into AC outlet provides 12 volts AC at ½ amp by two screw terminals. Great for various clocks, chargers, adding machines, etc. New

$2.50 ea. 5/$11.00

POWER SUPPLY

LAMBDA 5VDC 74 AMP

LV-EE-5-OV $125.00

NJE 5/OUP-D5

5 VDC 32 AMP $75.00

CLOCK KIT $14.00

Includes all parts with MM5316 chip, etched & drilled PC board, transformer, everything except case.

SP-284 $14.00 each 2/$25.00



ASCII KEYBOARD, brand new w/TI ASCII chip inplace & data $45.00

COMPUTER GRADE LOGIC SUPPLY CAPS, BRAND NEW

47,000 Uf	25V	$2.00	ST	1,000	50	.90	AL
32,000	25	1.75	ST	3,300	35	1.25	AL
160,000	10	2.00	ST	1,600	20	.60	AL
66,000	10	2.00	ST	8,000	16	1.25	AL
1,000	60	.90	AL	500	6	.35	AL
2,000	55	1.00	AL	"ST" screw top .... "AL" axial			

5 VOLT 1 AMP REGULATED power supply kit for logic work. All parts including LM-309K $7.50

DUMMY LOAD RESISTOR, non inductive, 50 ohm 5 watt $1.00

"AA" NICAD CELLS brand new, fine biz for handy talkies $1.25 ea. 9/$9.00

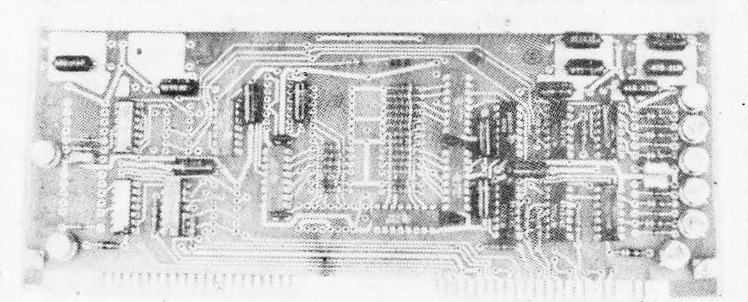
LINEAR by RCA, brand new, gold bond process

301	$ .60	747	$ .82	MM5314	$3.00
307	.52	748	.50	MM5316	3.00
324	1.80	1458	.96	7001	8.00
339A	1.60	3401	.80		
741	.50	555 timer	.60		

MEMORY SYSTEM $125.00

New memory system by Honeywell, small ... measures only 9x4x1 inches. 1024 core memory, 1024 words with 8,9,10 bits/word. Random access, with all logic, register, timing, control, core select and sense functions in one package. New, booklet of schematics and data. Looks like a good beginning for a mini-computer. Limited supply on hand.

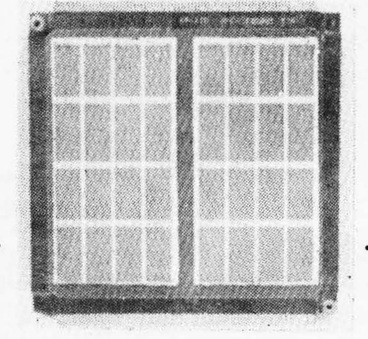
Ship wgt 3 lbs. #SP-79 .... $125.00



CORE MEMORY

Another brand new memory, ultra small. Measures only 4 x 4 inches with format on one plane of 32 x 32 x 16 (16,384). Only about 35 units of this on hand.

#SP-81 .... $20.00



Meshna

FREE CATALOG SP-7 NOW READY

Please add shipping cost on above.

MESHNA PO Bx 62 E. Lynn Mass. 01904

BYTE reader service

To get further information on the products advertised in BYTE, fill out the reader service card with your name and address. Then circle the appropriate numbers for the advertisers you select from this list. Add a 9 cent stamp to the card, then drop it in the mail. Not only do you gain information, but our advertisers are encouraged to use the marketplace provided by BYTE. This helps us bring you a bigger BYTE.

***Reader service inquiries not solicited. Correspond directly with company.**

BOMB: BYTE's Ongoing Monitor Box

BYTE would like to know how readers evaluate the efforts of the authors whose blood, sweat, twisted typewriter keys, smoking ICs and esoteric software abstractions are reflected in these pages. BYTE will pay a $50 bonus to the author who receives the most points in this survey each month.

Page No.	Article	LIKED LEAST … BEST
8	Lehman: Small Business Accounting System	0 1 2 3 4 5 6 7 8 9 10
16	Gantt: Build a Television Display	0 1 2 3 4 5 6 7 8 9 10
22	Herman: Programming for the Beginner	0 1 2 3 4 5 6 7 8 9 10
28	Lett: High School Computer System	0 1 2 3 4 5 6 7 8 9 10
32	Suding: Systems Approach to a Personal uP	0 1 2 3 4 5 6 7 8 9 10
40	Abbott: Building an M6800 Microcomputer	0 1 2 3 4 5 6 7 8 9 10
48	Hansford: Strike a MATCH	0 1 2 3 4 5 6 7 8 9 10
58	Nelson: "Chip" Off the Olde PDP 8/E, Part 2	0 1 2 3 4 5 6 7 8 9 10
66	Gable: Interact With an ELM	0 1 2 3 4 5 6 7 8 9 10
76	Zarrella: Introduction to Addressing Methods	0 1 2 3 4 5 6 7 8 9 10
82	Wadsworth: SCELBAL	0 1 2 3 4 5 6 7 8 9 10

March BOMB Winner

Winner of the $50 prize for the most popular article in the March 1976 BYTE is Jack Hemenway's "The COMPLEAT Tape Cassette Interface." A close second was Don Lancaster's "Build the BIT BOFFER." In third place was William Manly's "Magnetic Recording for Computers." The deadline for receipt of June BOMB evaluations is July 16, 1976.■

Continued from page 4

Given this goal of maximizing sales, what better way to do that than to maximize the utility of the product to its users by engineering it to possess the greatest generality? For the main frame kit manufacturers, this means that providing the standards capability allows the customer to interface a wide range of specialized applications oriented peripherals with a minimum of trouble. For the peripherals manufacturer, this allows the product to be sold to the owners of all the central processors which provide the standard interface, thus ensuring the widest possible market.

The Standard – A Summary

As the goal of parallel interface standards activities, there are three major technical points to consider:

Logical Definitions: The standard should define the data, control and addressing lines which are part of the interface. This definition would also include recommended sequences for such common operations as input data transfer, output data transfer, interrupt handshaking, etc.

Electronic Definitions: The standard should specify the physical parameters of the interface: logic level voltages, drive capacity, etc.

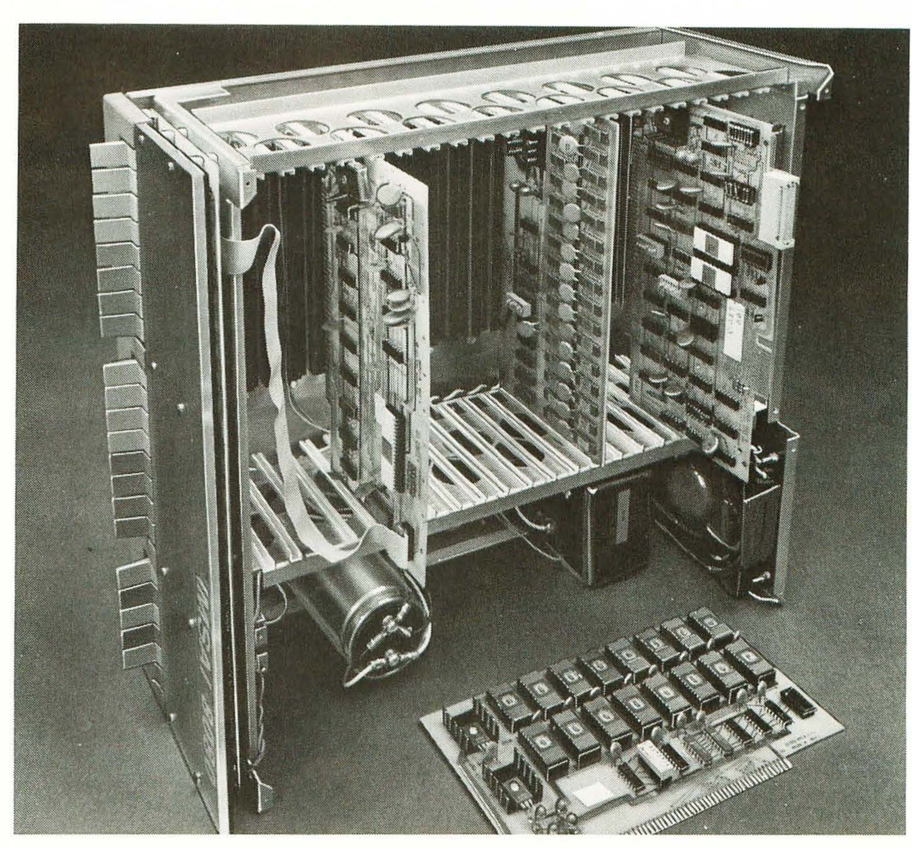
Physical Definitions: The logical definitions should be associated with the pinouts of one or more "recommended" connectors. Connector choices specified in the standard will help make it a more usable definition by limiting the number of possible alternatives.

An important point to remember is that the purpose of the standard is to create a definition which is widely publicized and can be used as a reference point by engineers and users of the equipment. With a standard, variations from its definition can be fully documented without ambiguity. (I owe this point to Calvin Moerrs of Rockford Research Inc in conversation at the March 1976 New England Computer Society meeting.)

To help encourage work on standards for the personal computing field, BYTE is organizing a technical session to be held at the Personal Computing '76 show in Atlantic City NJ August 28 and 29. A full page description of the standards session activity is found on page 5 of this issue. The parallel interface standard activities will form one of three areas of technical discussion identified at the time of this writing (March 26 1976).■

If you thought a rugged, professional yet affordable computer didn't exist,

think IMSAI 8080.



Sure there are other commercial, high-quality computers that can perform like the 8080. But their prices are 5 times as high. There is a rugged, reliable, industrial computer, with high commercial-type performance. And prices that are competitive with Altair's hobbyist kit. The IMSAI 8080. Fully assembled, it's $931. Unassembled, it's $599. And ours is available now.

In our case, you can tell a computer by its cabinet. The IMSAI 8080 is made for commercial users. And it looks it. Inside and out! The cabinet is attractive, heavy-gauge aluminum. The heavy-duty lucite front panel has an extra 8 program controlled LED's. It plugs directly into the Mother Board without a wire harness. And rugged commercial grade paddle switches that are backed up by reliable debouncing circuits. But higher aesthetics on the outside is only the beginning. The guts of the IMSAI 8080 is where its true beauty lies.

The 8080 is optionally expandable to a substantial system with 22 card slots in a single printed circuit board. And the durable card cage is made of commercial-grade anodized aluminum. The Altair kit only provides 16 slots maximum in four separate sections, each section requiring 200 solder connections.

The IMSAI 8080 power supply produces a true 28 amp current, enough to power a full system. The Altair produces only 8 amps.

You can expand to a powerful system with 64K of memory, plus a floppy disk controller, with its own on board 8080 – and a DOS. An audio tape cassette input device, a printer, plus a video terminal and a teleprinter. These peripherals will function with an 8-level priority interrupt system. IMSAI BASIC software is available in 4K, 8K and 12K, that you can get in PROM. And a new $139 4K RAM board with software memory protect.

Find out more about the computer you thought didn't exist. Get a complete illustrated brochure describing the IMSAI 8080, options, peripherals, software, prices and specifications. Send one dollar to cover handling to IMS. The IMSAI 8080. From the same technology that developed the HYPERCUBE Computer architecture and Intelligent Disk systems.

Dealer inquiries invited.

IMSAI 8080

IMS

IMS Associates, Inc. **Dept. B-6**
14860 Wicks Boulevard
San Leandro, CA 94577
(415) 483-2093


MITS ALTAIR 680
MITS ALTAIR 8800 COMPUTER